(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
31 October 2002 (31.10.2002)

PCT

(10) International Publication Number
WO 02/086718 A1

(51) International Patent Classification[7]: G06F 11/00, 13/00, H04L 9/00, 9/08, 9/32

(21) International Application Number: PCT/US02/12470

(22) International Filing Date: 18 April 2002 (18.04.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/284,914 18 April 2001 (18.04.2001) US
10/117,868 5 April 2002 (05.04.2002) US

(71) Applicant *(for all designated States except US)*: **IPASS, INC.** [US/US]; 3800 Bridge Parkway, Redwood City, CA 94065 (US).

(72) Inventors; and

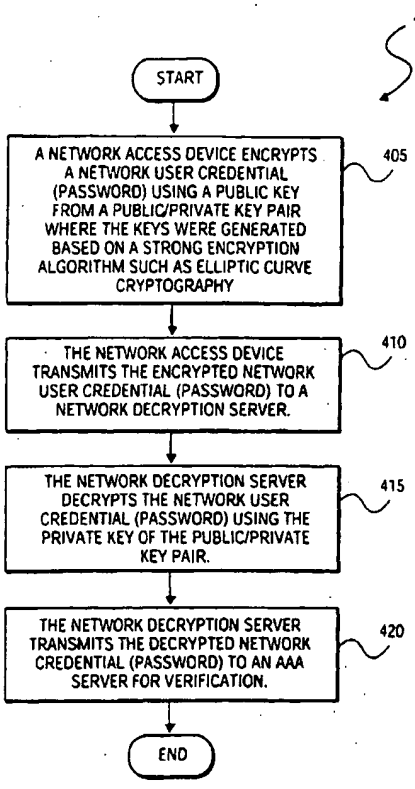
(75) Inventors/Applicants *(for US only)*: **ALBERT, Roy, David** [US/US]; 6529 Fall River Drive, San Jose, CA 95120 (US). **EDGETT, Jeff, Steven** [US/US]; 151 S. Bernardo #24, Sunnyvale, CA 94086 (US). **SUNDER, Singam** [IN/US]; 539 Isaac Court, San Jose, CA 95136 (US). **UNDERWOOD, Jim** [US/US]; 6896 Corte Sonada, Pleasanton, CA 94566 (US).

(74) Agents: **MALLIE, Michael, J.** et al.; Blakely, Sokoloff, Taylor & Zafman LLP, 7th floor, 12400 Wilshire Boulevard, Los Angeles, CA 90025 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG,

[Continued on next page]

(54) Title: METHOD AND SYSTEM FOR SECURELY AUTHENTICATING NETWORK ACCESS CREDENTIALS FOR USERS



(57) Abstract: A methods is provided to secrurely authenticate user credentials. The method includes encrypting a user credential with a public key at an access device wherein the public key is part of a public/private key pair suitable for use with an encryption algorithm (405). The encrypted network user credential is transmitted from the access device to a decryption server where it is decrypted with a private key, the private key being part of the public/private key pair suitable for use with the encryption algorithm. The decrypted user credential is then transmitted from the decryption server to an authentication server for verification (420). The decryption server typically forms part of a multi-party service access environment including a plurality of access providers, the method including decrypting the user credential of a user proximate an access provider associated with the user credential. The method can be used in legacy protocols such as Point-to-point protocol (PPP), Password Authentication Protocol (PAP), Challenge-Handshake Authentication Protocol (CHAP), Remote Authentication Dial In User Service (RADIUS) protocol, Terminal Access Controller Access Control System (TACACS) protocol, Lightweight Directory Access Protocol (LDAP), NT Domain authentication protocol, Unix password authentication protocol, HyperText Transfer Protocol (HTTP), HyperText Transfer Protocol over Secure sockets layer (HTTPS), Extended Authentication Protocol (EAP), Transport Layer Security (TLS) protocol, Token Ring protocol and/or Secure Remote Password protocol (SRP).

WO 02/086718 A1

SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZM, ZW.

(84) **Designated States** *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*
— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

METHOD AND SYSTEM FOR SECURELY AUTHENTICATING NETWORK ACCESS CREDENTIALS FOR USERS

[0001] The present application claims the benefit of the filing date of US provisional patent application no. 60/284914 entitled "METHOD FOR ASSURED PASSWORD SECURITY WHEN USING INSECURE FACILITIES" filed April 18, 2001.

FIELD OF THE INVENTION

[0002] The present invention relates generally to computer network security. More particularly, the present invention relates to protecting computer networks against unauthorized access and to a method and system to securely authenticate network access credentials for users.

BACKGROUND

[0003] Within the past decade, the number of users accessing computer networks such as the Internet has exploded. Typically, users access the Internet through an Internet Service Provider (ISP). A network user attempting to gain access to the Internet or a corporate local area network (LAN) must generally enter a username and password for identification verification purposes. A problem with this process is that the password is generally not secure when transmitted to the ISP using many standard authentication protocols.

[0004] Figure 1 illustrates a diagram of a prior art ISP network configuration 100 in which network user credentials are authenticated using an insecure method. An ISP network 145 includes a network access server (NAS) 120 connected to a modem pool 115 and to the Internet 150 via a gateway 125. The ISP network 145 is also connected to an authentication server (AAA server) 135. The AAA server 135 may be local to the ISP network 145 or in a remote location a great distance from the ISP Network 145.

[0005] To establish an Internet connection, a network user typically executes a dial-up networking application on a network access device 105. The dial-up networking application prompts the network user to enter a network username and a network password and manipulates a modem 110 in order to initiate a modem session with the modem pool 115 over a public switched telephone

network (PSTN) 140. After a modem session has been established, the dial-up networking application begins communicating with the NAS 120 for purposes of establishing a data connection and authenticating the network user.

[0006] One of the more common data communication protocols used to establish connections between computers is the point-to-point protocol (PPP). One particularly well-known authentication protocol, which is commonly used in conjunction with PPP, is the Password Authentication Protocol (PAP). A dial-up networking application configured to use PAP repeatedly sends the username and password pair over the established data connection until an authorization acknowledgement signals is received or the connection is terminated. The dial-up networking application is configured to control the frequency and timing of the username and password transmission.

[0007] A problem with PAP is that the password is not encrypted before it is sent over the data connection, but instead, it is sent as clear text. This means that the password is susceptible to interception by a hacker. For example, a hacker with access to the data connection can use a network monitoring application to capture and display data packets that are sent across the data connection. Such network monitoring applications are common and are often referred to as packet sniffing or packet snooping applications due to their illicit use.

[0008] Referring again to Figure 1, once the username and password pair is received at the NAS 120, Remote Authentication Dial In User Service (RADIUS), another standard authentication protocol, is typically used to transmit the network username and password pair to an ISP authentication system 155. The RADIUS protocol provides for the symmetric encryption of the password before it is sent to the AAA server 135 in the ISP authentication system 155. The encryption method is considered symmetric because the NAS 120 and the AAA server 135 share a secret key used in the encryption algorithm. The NAS 120 uses the secret key to "lock", or encrypt, the password, while the AAA server 135 uses the secret key to "unlock", or decrypt, the password before checking the password against the password stored in an authentication database 130.

[0009] A problem with the RADIUS symmetric encryption method is that it is susceptible to a form of attack known as a "dictionary" attack. In a dictionary attack, a hacker with knowledge of the encryption algorithm intercepts an encrypted password with a packet sniffing application. Then, the hacker repeatedly tries a series of keys until one is found that yields readable characters. To make matters worse, once the secret key is compromised, a hacker can readily decrypt any password intercepted between the NAS 120 and the AAA server 135.

[00010] In response to the weaknesses inherent in the PAP/RADIUS authentication method just described, the Challenge Handshake Authentication Protocol (CHAP) was developed. In a system implemented to use CHAP, the dial-up application in the network access device 105 negotiates with the NAS 120 to use CHAP as the authentication protocol, instead of PAP. Next, the NAS 120 generates a random number and sends it to the network access device 105. The dial-up networking application executing on the network access device 105 uses the random number to generate a non-reversible hash of the password, which is then sent to the NAS 120. The NAS 120 then uses the RADIUS protocol and sends the non-reversible hash and the random number used to generate the hash to the AAA server 135. The AAA server 135 retrieves the clear text password from the authentication database 130 and repeats the hash operation using the random number received from the NAS 120. Finally, the AAA server 135 compares its generated hash value with the hash value received from the NAS 120. If the hash values are the same, the authentication is considered successful and the AAA server 135 sends the appropriate acknowledgement signal to the network access device 105.

[00011] A problem with the CHAP/RADIUS method for user authentication is that all three systems, namely the network access device 105, the NAS 120 and the AAA server 135, must be configured to use CHAP in order to take advantage of the added security. If any of the three are not configured to use CHAP, the dial-up networking application on the network access device 105

uses the PPP protocol to negotiate with the NAS 120 to use PAP as the authentication protocol.

[00012] Another disadvantage of using the CHAP/RADIUS method is that in order for CHAP to be implemented properly, the AAA server 135 must have access to clear text passwords. Many authentication systems do not store passwords in clear text form because of the added security risk that would result if the system were compromised and the passwords stolen.

[00013] More recently, authentication systems have deployed an authentication protocol referred to as Extensible Authentication Protocol (EAP). EAP works in much the same way as CHAP, except that the AAA server 135, not the NAS 120, generates the random number which the network access device 105 uses to hash the password. Consequently, EAP is subject to the same disadvantages of CHAP. Particularly, EAP is only effective if all systems in the authentication chain employ EAP.

[00014] With the advent of Broadband access, both wireless and wireline (ethernet) access providers employ web browser based authentication systems. The web browser uses Hyper Text Transport Protocol (HTTP) or Hyper Text Transport Protocol over Secure sockets layer (HTTPS) for transmitting the user credentials to the access point. A problem with HTTP is that the password is not encrypted before it is sent over the data connection, but instead, it is sent as clear text. This means that the password is susceptible to interception by a hacker. For example, a hacker with access to the data connection can use a network monitoring application to capture and display data packets that are sent across the data connection. Such network monitoring applications are common and are often referred to as packet sniffing or packet snooping applications due to their illicit use. A problem with HTTPS is that the access point needs to obtain the certificate from a well-known Certificate Authority (CA). This increases the cost of setting up the access point. The strength of the encryption used by HTTPS is regulated by government export restrictions. The web browsers include the weaker keys by default, and the users are expected to upgrade the encryption strength depending upon export restrictions. For the

purposes of this specification, the term "connection application" should be construed as including, but not limited to, any device (both hardware and software) including functionality to authenticate data e.g., a peer-to-peer authentication arrangement, a dialer, a smart client, a browser, a supplicant, a smart card, a token card, a PDA connection application, a wireless connection, an embedded authentication client, an Ethernet connection, or the like.

SUMMARY OF THE INVENTION

[00015] In accordance with the invention, there is provided a method to securely authenticate user credentials, the method including:

encrypting a user credential with a public key at an access device, the public key being part of a public/private key pair suitable for use with an encryption algorithm;

transmitting the encrypted network user credential from the access device to a decryption server;

decrypting the user credential at the decryption server with a private key, the private key being part of the public/private key pair suitable for use with the encryption algorithm; and

transmitting the decrypted user credential from the decryption server to an authentication server for verification.

[00016] Further in accordance with the invention, there is provided a method of authenticating user data of a user requesting access to a service access system including a plurality of service providers, the method including:

encrypting the user data with a public key, the public key being part of a public/private key pair suitable for use with an encryption algorithm; and

transmitting the encrypted user data to a decryption server for decryption using the private key.

[00017] Still further in accordance with the invention, there is provided a method of authenticating user data of a user requesting access to a service access system including a plurality of service providers, the method including:

receiving encrypted user data from an access device;

decrypting the encrypted user data using a private key; and

transmitting the decrypted user data to an authentication server for authentication.

[00018] The invention extends to a computer readable medium, having instructions stored thereon to execute any of the described therein methods.

[00019] In accordance with a further aspect of the invention, there is provided a computer to authenticate user data of a user requesting access to a service access system including a plurality of service providers, the computer including:

a receiver to receive encrypted user data from an access device;

decryptor to decrypt the encrypted user data using a private key; and

a transmitter to transmit the decrypted user data to an authentication server for authentication.

[00020] Other features and advantages of the present invention will be apparent from the drawings and detailed description that follow.

BRIEF DESCRIPTION OF THE DRAWINGS

[00021] The present invention is illustrated by way of example, and not intended to be limited by the figures of the accompanying drawings, in which like references indicate the same or similar elements and in which:

[00022] Figure 1 is a diagram of a prior art ISP network configuration in which network user credentials are authenticated using an insecure method;

[00023] Figure 2 is a diagram of a network configuration including an ISP network, a network access device, and a network decryption server, consistent with an embodiment of the invention;

[00024] Figure 3 is a diagram of a network configuration including a remote ISP network, a network access device and a network decryption server, consistent with an embodiment of the invention;

[00025] Figure 4 is an exemplary flow diagram of the operations for a method to securely authenticate network user credentials;

[00026] Figure 5 is a block diagram of a multi-party service access environment, in accordance with an exemplary embodiment of the invention,

which includes a number of service providers, an access broker system and multiple customers;

[00027] Figure 6 is a schematic diagram illustrating operation of an access broker system, in accordance with an exemplary embodiment of the invention, to provide roaming Internet access;

[00028] Figure 7 is a schematic functional block diagram of a connect dialer in accordance with a exemplary embodiment of the invention;

[00029] Figure 8 is a schematic functional block diagram of a transaction server, in accordance with an embodiment of the invention, which includes decryption functionality;

[00030] Figure 9 is a schematic functional block diagram of customer or roam server, in accordance with another embodiment of the invention, which includes decryption functionality;

[00031] Figure 10 is a schematic flow diagram of an exemplary encryption method performed by the connect dialer;

[00032] Figure 11 is a schematic flow diagram of an exemplary decryption method performed by the transaction server or customer server;

[00033] Figure 12 is a schematic flow diagram of an exemplary encryption method of checksum data;

[00034] Figure 13 is a schematic diagram of a computer system, which may be configured as a network access device or a network decryption server.

[00035] Figure 14 is a schematic block diagram illustrating operation of an access broker system to provide roaming Internet access, in accordance with one embodiment of the invention;

[00036] Figure 15 is a schematic block diagram of exemplary physical architecture of the access broker system of Figure 14;

[00037] Figure 16 is a schematic block diagram of an exemplary settlement system;

[00038] Figure 17A shows an exemplary data model used in the access broker system;

[00039] Figure 17B is a schematic diagram illustrating processing, in accordance with the invention, using a unique session identification also in accordance with the invention;

[00040] Figure 18 shows an exemplary unique session identification in accordance with one embodiment of the invention;

[00041] Figure 19 shows a schematic flow chart of methodology to identify missing transaction data records using a unique session identification; and

[00042] Figure 20 shows a schematic flow chart of unique session identification methodology at a connection application also in accordance with an embodiment of the invention.

DETAILED DESCRIPTION

[00043] A method and system for securely authenticating network user credentials or user data are described. A network access device encrypts a network user credential, such as a password, input by a network user. The network access device encrypts the network user credential with a public key, which is part of a public/private key pair, generated with a strong encryption algorithm. The network access device transmits the encrypted network password to a network decryption server. The network decryption server decrypts the network user credential using the private key of the public/private key pair. The network decryption server transmits the decrypted password to an authentication (AAA) server for verification. If the password is positively verified at the AAA server, the AAA server sends an appropriate acknowledgment signal to the network access device indicating that the password has been properly verified or authenticated. Based on the acknowledgement signal, the network access device gains access to the Internet or some other resource.

[00044] By encrypting the network password at the network access device with an asymmetric public key based on a strong encryption algorithm, the password can be securely transmitted from the network access device to a network decryption server. If the encrypted password is captured by a sniffing or snooping application at some point between the network access device and

the network decryption server, the encrypted password can only be decrypted with knowledge of the correct private key and the encryption algorithm. Preferably, decryption of the user credentials takes place as close as possible to the source which the user wishes to access.

[00045] The embodiment of the invention depicted in the drawings is independent of the underlying authentication protocols and therefore can be implemented to work with a variety of new and existing authentication protocols. Moreover, this embodiment of the invention provides for secure authentication while resolving the need to fully standardize the capability of the authentication chain. For example, by passing encrypted data through standard PPP/RADIUS information fields, the invention provides a secure authentication method without the hassle and expense of implementing and configuring network equipment to work with more complex authentication protocols, such as CHAP and EAP. It is, however, to be appreciated that the invention may be used with CHAP, EAP and other protocols and is not limited to application in a PAP/RADIUS environment.

[00046] Figure 2 is a diagram of a network configuration 200 including an ISP network 255, a network access device 205 and a network decryption server 240, consistent with one embodiment of the invention. The ISP network 255 includes a NAS 220, a modem pool 215 and a gateway 225. The ISP network 255 is connected to the Internet via the gateway 225 and connected to an ISP authentication system 265 via a connection between the NAS 220 and a network decryption server 240. In one embodiment, the ISP network 255 and the ISP authentication system 265 are physically located within the same facility. However, in an alternative embodiment, the ISP authentication system 265 is located in one facility and connected via a wide area network (WAN) to one or more ISP networks, such as the ISP network 255. This type of configuration allows for the individual ISP networks to be strategically located in different geographical areas thereby allowing customers to access the network via a local telephone call, while centralizing the authentication system for added security.

[00047] In one embodiment of the invention, to access the Internet 260, a network user executes a dial-up connection application on the network access device 205. In alternative embodiments, other types of network connection applications may be utilized to access the Internet. The dial-up connection application prompts the network user to input a network username and a network password and manipulate a modem 210, causing it to establish an audio communication session with the modem pool 215. Although the modem 210 is shown in Figure 2 as an external device, in alternative embodiments of the invention, the modem 210 may be an internal device integrated with the network access device 205. Once an audio communication session has been established, the NAS 220 begins communicating with the network access device 205 for the purpose of authenticating the network user.

[00048] Before the network access device 205 sends the network credentials entered by the network user, the network password is encrypted. The password is encrypted using the public key of a public/private key pair. This encryption technique is well known in the art and is generally referred to as asymmetric public key cryptography. In asymmetric public key cryptography, a person makes one key publicly available and holds a second, private key. A message is "locked", or encrypted, with the public key, sent, and then "unlocked", or decrypted, with the private key.

[00049] In the embodiment of the invention depicted in the drawings, a strong encryption algorithm is used to generate the public/private key pair. The public key and private key have a mathematical relationship based on an elliptic curve. This encryption technique is well known in the art and is generally referred to as elliptic curve cryptography or ECC. Public key encryption algorithms rely on a one-way mathematical problem, which makes it easy to generate a public key from a private key but difficult to deduce the private key, given the public key. Elliptic curve systems use an algebraic formula to determine the relationship between public and private keys within the universe created by an elliptic curve. Elliptic curve cryptography is advantageous because the key sizes are small relative to other strong

encryption techniques. This allows a password to be encrypted with strong encryption and yet, an encrypted password still fits in the password data field defined by the popular authentication protocols, such as PAP, CHAP, EAP, and RADIUS.

[00050] Referring again to Figure 2, the public key is known to the network access device 205, while the private key is stored in a private key database 245. The network access device 205 encrypts the password using the public key before sending the network username and the encrypted network password to the NAS 220. The NAS 220 forwards the network username and the encrypted network password to the network decryption server 240. The network decryption server 240 uses the network username as an index into the private key database 245 and retrieves the private key associated with the network username. Then, the network decryption server 240 uses the private key to decrypt the encrypted network password and to generate the original clear text password as input by the network user.

[00051] Finally, the network decryption server 240 forwards the network username and the clear text network password to the AAA server 235 for verification. The AAA server 235 uses the network username as an index into the authentication database 230 to retrieve the official password that is associated with the network username. If the official password matches the password input by the network user and sent by the network access device 205, the AAA server 235 sends an appropriate acknowledgment signal to the NAS 220, and the NAS 220 forwards the signal to the network access device 205, acknowledging the successful verification and granting access to the Internet or some other resource.

[00052] One embodiment of the invention is independent of the authentication protocols used to send the credentials from the network access device 205 to the NAS 220 and ultimately to the AAA server 235. For example, the invention can be implemented to work with popular authentication protocols such as PAP, CHAP, EAP and RADIUS, among others.

[00053] For one embodiment of the invention, the NAS 220 is configured to use PAP and RADIUS for authenticating network user credentials. When configured for PAP/RADIUS, the NAS 220 negotiates the use of PAP with the network access device 205 when the communication session between the NAS 220 and the network access device 205 is initiated. The NAS 220 is configured as a RADIUS client of the AAA server 235, which is a RADIUS server. The network decryption server 240 is also configured as a RADIUS server, but acts as a RADIUS proxy client to the AAA server 235. In this configuration, the network access device 205 encrypts the password, as entered by the network user. Then, the network access device 205 creates a PAP packet and places the network username and encrypted network password into the proper fields within the packet. Next, the network access device 205 sends the PAP packet to the NAS 220. The NAS 220 forwards the data to the network decryption server 240 using a RADIUS packet. The network decryption server 240 decrypts the password and uses RADIUS to forward the clear text password to the AAA server 235 for verification.

[00054] In an alternative embodiment, the NAS 220 is configured to use CHAP and RADIUS to authenticate network user credentials. In a network configured to use CHAP/RADIUS, the NAS 220 negotiates with the network access device 205 to use CHAP as the authentication protocol, instead of PAP. Next, the NAS 220 generates a random number and sends it to the network access device 205. The dial-up connection application executing on the network access device 205 uses the random number to generate a non-reversible hash of the password using a pre-determined encryption algorithm. Rather than encrypt the actual password, the network access device 205 encrypts the non-reversible hash of the network password in accordance with the exemplary embodiment of the invention as described above. The network access device 205 creates a CHAP packet and sends the network username and the encrypted non-reversible hash to the NAS 220.

[00055] The NAS 220 sends the data, including the network username, the encrypted non-reversible hash, and the original random number used to

generate the non-reversible hash, to the network decryption server 240 using the RADIUS protocol. The network decryption server 240 decrypts the non-reversible hash and replaces the non-reversible hash in the RADIUS packet, which is forwarded to the AAA server 235.

[00056] The AAA server 235 receives the packet and retrieves the password associated with the network username from the authentication database 230. The AAA server 235 uses the random number originally generated at the NAS 220 to perform a hash operation on the original password retrieved from the authentication database 230. Next, the AAA server 235 compares the hash it generated to the hash it received from the network access device 205. If the two hashes match, the verification is successful and the AAA server 235 sends an appropriate acknowledgment signal to the network access device 205 granting access to the Internet 260 or some other resource.

[00057] In another embodiment of the invention, the NAS 220 is configured to use EAP and RADIUS. EAP works in much the same way as CHAP, except the random number sent to the network access device 205 is generated by the AAA server 235 instead of the NAS 220. Because the invention works with any authentication protocol, the invention can easily be implemented to work with a variety of network configurations and provides a very strong, minimal level of security using LEGACY systems.

[00058] Figure 3 is a diagram of a network configuration 300 including a remote ISP network 365, a network access device 305 and a network decryption server 350, consistent with one embodiment of the invention. The remote ISP network 365 includes a NAS 320, a modem pool 315 and a gateway 325. The remote ISP network 365 is connected to the Internet 370 via the gateway 325 and connected to a remote ISP authentication system 375 via a connection between the NAS 320 and the AAA server 335. The remote ISP authentication system 375 is connected to a local ISP authentication system 380 via a WAN connection between the AAA server 335 and the network decryption server 350.

[00059] The configuration 300 allows a network user via the network access device 305 to accesses the Internet 370 through the remote ISP network 365. A

local ISP, which operates and maintains the local ISP authentication system 380, makes arrangements with a remote ISP, such that network users of the local ISP are allowed access to the Internet via the remote ISP network 365, which is maintained and operated by the remote ISP. This type of business arrangement might exist where the remote ISP is located in a distant geographical area or different country from the local ISP. The embodiment of the invention depicted in Figure 3 is particularly advantageous in this type of configuration because of the inability of the local ISP operators to control who has access to the equipment that comprises the remote ISP network 365 and the remote ISP authentication system 375. Further, the remote ISP network 365 only has access to an encrypted version of the password thereby enhancing security.

[00060] The embodiment of the invention illustrated in Figure 3 works in much the same way as discussed above in relation to Figure 2, except that the encrypted password passes through the remote ISP network 365 and the remote ISP authentication system 375. Referring to Figure 3, to access the Internet 370, a network user executes a dial-up connection application on the network access device 305. The dial-up connection application prompts the network user to input a network username and network password and manipulates the modem 310, causing it to establish an audio communication session with the modem pool 315. Once an audio communication session has been established, the NAS 320 begins communicating with network access device 305 for the purpose of authenticating the network user.

[00061] Before transmitting the network password to the NAS 320, the network access device 305 encrypts the network password with a public key as discussed above. The network access device 305 then creates a data packet destined for the local ISP authentication system 380 and forwards the packet to the NAS 320 of the remote ISP 365. The NAS 320 receives the data packet containing the encrypted password and forwards it to the remote ISP authentication system 375 and the AAA server 335 in particular. The AAA server 335 examines the data packet, discovers it is destined for the local ISP

authentication system 380, and forwards the data packet to the network decryption server 350.

[00062] The network decryption server 350 receives the data packet and retrieves the private key associated with the network username from a private key database 355. Then, the network decryption server 350 decrypts the encrypted password and forwards the data packet with the clear text password to the AAA server 345 for verification. The AAA server 345 uses the network username as an index into the authentication database 340 to retrieve the clear text password associated with the username from the authentication database 340. If the retrieved password matches the password received from the network access device 305, then the AAA server 345 sends an appropriate acknowledgment signal to the AAA server 335 of the remote ISP 365. The AAA server 335 forwards the signal to the NAS 320. The NAS 320 forwards the signal to the network access device 305 acknowledging the successful verification and granting access to the Internet or some other resource. Thus, decryption takes place in proximity to a local ISP associated with the user and any one or more intermediary ISPs only have access to encrypted authentication data.

[00063] Figure 4 illustrates a flow diagram of the operations 400 for a method to securely authenticate network user credentials, consistent with one embodiment of the invention. The method begins at operation 405. At operation 405, a network access device uses a public key, which is part of a public/private key pair, to encrypt a network credential such as a password. The public/private key pair has been previously generated based on a strong encryption algorithm, such as elliptic curve cryptography.

[00064] At operation 410, the network access device transmits the encrypted network credential to a network decryption server. The encrypted password may be forwarded through several network nodes, including network access servers and AAA servers before it ultimately reaches the network decryption server.

[00065] At operation 415, the network decryption server decrypts the encrypted network credential using the private key of the public/private key

pair referred to above. The network decryption server may retrieve the private key from a private key database, using the username as an index into the private key database.

[00066] Finally, at operation 420, the network decryption server transmits the decrypted network credential to an AAA server for verification. The decrypted network credential may be forwarded over several network nodes, such as network access servers or other AAA servers before ultimately reaching the AAA server for verification.

[00067] A typical application of the invention is in a multi-party service access environment and its application therein is described below. Such applications typically include roaming users, multiple service providers and multiple customers. Further, such applications typically use PAP, CHAP, EAP, RADIUS or the like protocols which communicate user credentials in an insecure fashion. However, the embodiment described below allows secure authentication in LEGACY systems.

Terminology

[00068] For the purposes of the present specification, the term "service access transaction" includes any transaction between a service customer and a service provider for a user session. An example of such a service may be access to any communications network via any medium or protocol. For example, the communications networks may comprise packet-switched networks, circuit-switched networks, cable networks, satellite networks, terrestrial networks, wired networks, or wireless networks. The term "service access transaction", however, is not limited to a network access transaction, and encompasses a transaction pertaining to access to any one of a number of other services such as content, commerce and communications services.

[00069] For the purposes of the present specification, the term "customer" includes any entity involved in the purchase and/or consumption of service access, regardless of whether the service access is performed by the customer or not. For example, a "customer" may be an end-user consumer that actually

utilizes the service access, or a corporate entity to which such an end-user belongs, an Internet service provider, an Internet carrier, a reseller, or a channel.

Multi-party services access environment

[00070] This embodiment of the present invention discloses a multi-party access broker and settlement system for service access (e.g., Internet access, content access, commerce access, or communications access) services that enable a service provider (e.g., an ISP, a wireless service provider, a VPN service provider, a content distribution service provider, an e-commerce service provider or an application service provider) to offer relatively secure service access in a multi-party access environment using standard communication protocols (e.g., PPP, HTTP) and standard authentication protocols (e.g., RADIUS, PAP, EAP or the like). Such protocols typically define a user field of a maximum length and the exemplary embodiment of the invention describes, inter alia, a method and system to provide secure authentication within a field with the abovementioned maximum length. Accordingly, the invention may be applied to LEGACY systems.

Overview

[00071] Figure 5 is a block diagram of an exemplary multi-party service access environment 450, in the exemplary form of a network access environment, which includes a number of service providers 452, an access broker system 454, and multiple customers (or consumers) 456. At a high level, the service providers 452 have service (e.g., access, content, e-commerce services etc.) capacity that is sold, via the access broker system 454, to the multiple customers 456. Accordingly, the access broker system 454 may be regarded as purchasing service capacity (e.g., service access), which is then resold to the customers 456. While the service to which access is provided below is network access, it will be appreciated that access is described below as an exemplary service and, for the purposes of this specification should be taken to include any form of access as described above. In the exemplary embodiment, the service providers 452 may include any communication network service providers, such as ISPs 458 (e.g., UUNet Technologies,

Genuity, CompuServe Network Services, EQUANT, Hong Kong Telecom, etc.), wireless access providers 460 (e.g., Verizon, Sprint, Pacific Bell), content distribution providers 462 and e-commerce providers 464. The service providers 452 may, however, include any number or type of service providers providing any number of services (e.g., access, content, communications or e-commerce services, to name but a few).

[00072] The exemplary access broker system 454 includes a number of components. A connection application is a client application typically in the form of a dial-up application or connect dialer 466, installed on a service or network access device (e.g., a computer system such as the access devices 205, 305 in Figures 2 and 3) of a customer 456 that facilitates convenient access to a communications network. In one embodiment, the connect dialer 466 may provide a simple point-and-click interface for dialing into a worldwide connection network of the access broker system 454. To this end, the connect dialer 466 may store multiple phone numbers for multiple ISPs worldwide with potentially different setup and dial-up scripting information. As described above broadly with respect Figures 1 to 4, the connect dialer 466 encrypts user data and counter data in such a fashion so that it may be included in the user string permitted or allowed by known protocols such as Point-to-Point protocol (PPP), Password Authentication Protocol (PAP), Challenge-Handshake Authentication Protocol (CHAP), Remote Authentication Dial In User Service (RADIUS) protocol, Terminal Access Controller Access Control System (TACACS) protocol, Lightweight Directory Access Protocol (LDAP), NT Domain authentication protocol, Unix password authentication protocol, HyperText Transfer Protocol (HTTP), HyperText Transfer Protocol over Secure sockets layer (HTTPS), Extended Authentication Protocol (EAP), Transport Layer Security (TLS) protocol, Token Ring protocol and/or Secure Remote Password protocol (SRP).

[00073] The environment 450 also includes a plurality of transaction servers 468 that provide trusted third-party functionality of routing and logging user identification information, authorization responses and usage, and accounting

information. The transaction servers 468 include decryption functionality and may thus define decryption servers.

[00074] Whereas the connect dialer 466 is installed on a client or user network access device 205, 305, the netservers 470 are installed at a "remote" ISP allowing its POPs to be utilized by roaming users, and roam servers 472 reside at a "home" ISP to allow a roam user access an associated home network. It should be noted that the transaction servers 468 operate to route messages between the network and roam servers 470 and 472.

[00075] A settlement system 474, including a flexible pricing engine 476, performs financial settlement of service access transactions between the service providers 452 and the customers 456. The access broker system 454 is also includes a Service Quality Monitor 478 (SQM) that facilitates the collection and analysis of quality of service (QoS) information for services provided to customers 456 and a phonebook management system 480 that facilitates management of multiple connect dialers 466 used by customers 456. The transaction servers 468 are accessed by the settlement system 474 to load transaction data. The various components in the environment 450 may include aspects of known functionality and, dependent upon the specific embodiment of the invention, certain components may be omitted.

The Customers

[00076] The customers 456, in the embodiment depicted in the drawings, are arranged in a multi-tier customer structure, whereby the access broker system 454 may interact with customers 456 that operate according to a variety of business plans and needs. At one end of the spectrum, the customer 456 may comprise an individual end-user that subscribes to a roaming system facilitated via the access broker system 454. Alternatively, the customer 456 may be in the form of a corporate customer 482 (e.g., a corporation or business) that purchases roaming Internet access for employees of the corporation.

[00077] Each customer 456 may also comprise an ISP customer 484 that purchases roaming Internet access for resale to its customers (e.g., end-users 486 and corporate customers 482). Each customer 456 may also operate as a

solution partner or reseller 488 that markets and resells roaming Internet access brokered by the access broker system 454 to end-users 486, corporate customers 482 and/or ISP customers 484.

[00078] The customers 456 may also include parties regarded as Internet Carriers 490 (e.g., IXCs, RBOs, CLECs, ILECs and ISPs). It will thus be appreciated that in the multi-party access environment 450 a number of different service providers may participate in providing access to a roaming user and, accordingly, customer security issues and, in particular, secure authentication of users, are of importance. Also, as the number of participants increases, accounting issues tend to become more complex.

Roaming Service Access

[00079] Referring in particular to Figure 6, reference numeral 500 generally indicates an example of how the access broker system 454 may provide roaming Internet access in a relatively secure manner in accordance with an embodiment of the invention. When a roaming user 502, shown to be a subscriber to a "home" ISP 504, connects to a remote ISP 506 that provides a local POP 508 within a specific geographic area 510, the roaming user 502 inputs the same user name 512 and password 514 (i.e., authentication data or user credentials) used when connecting via a POP 509 of the "home" ISP 504. However, standard or LEGACY multi-party access environments typically use PAP for dialup authentication and HTTP POST based authentication for wired and wireless broadband authentication. This results in the passwords being transported via insecure media and their confidentiality may be compromised and subsequently used to fraudulently access both networks of the access broker system 454 and the customers 456. In order to alleviate this problem, in accordance with one embodiment of the invention, user data is encrypted by the connect dialer 466 prior to communicating it to the POP 508, as described above with reference to Figures 1 to 4, and in the context of a multi-party environment with reference to Figures 5 to 13.

[00080] In the embodiment depicted in the drawings, the customers 456 use a web form for requesting the connect dialer 466. This web form may include

fields that can be used for specifying the required customizations. For example, the following fields are included in the web form for Secured Password Authentication in Plain-text (hereinafter referred to as "Secure PAP") -

Enable Secure PAP encryption: (Y/N)

Public Key: ****

Key Id: (0-9)

[00081] When a customer 456 wants to enable Secure PAP for their roaming users 502 (see Figure 6), they use an ECC utility that is included in their associated roam server 472. The roam server 472 runs an application supplied by the access broker system 454 to the customers 456 and generates a public/private key pair. The private key is typically added to an esp_key_pair.txt file. The public key is typically sent to the dialer support team of the access broker system 454 using an appropriate form. The dialer support team uses a dialer customization tool (DCT) to build the connect dialer 466 in accordance with one embodiment of the invention. The DCT tool includes a web page for specifying the encryption/decryption algorithm to be used and the ECC public/private keys.

[00082] The connect dialer 466 maintains a dialer id and counter (see block 520 in Figure 7) for generating a unique session id (see block 522) that uniquely identifies a user access session. The connect dialer 466 may, for example, obtain the dialer id from a web server of the access broker system 454 and, in use, the connect dialer 466 increments the counter for each dial attempt so that each user access session is uniquely identified. The dialer id and a value of the counter are used to create the unique session id prefix. In order to ensure the integrity of the dialer id and value or count of the counter (which are transmitted in the clear), these fields are used to generate a checksum character. The algorithm used for generating this checksum character is described in more detail below with reference to Figure 12. An exemplary embodiment of the unique session id is described in more detail later in this document.

[00083] The netserver 470 maintains a cache of authenticated user ids and passwords for a limited period so that subsequent authentications can be

performed from the cache (see block 538). Since the secure PAP changes the user id and password for each authentication, it breaks any caching feature at the netserver 470. Thus, in certain embodiments, in order to maintain compatibility with the standardized netserver cache, the dialer 466 may store a random point locally and reuse this for limited period of time (see block 540). After the aforementioned processing, the netserver 470 communicates the authentication data to the transaction server 468.

[00084] Referring in particular to Figure 8, reference numeral 550 generally indicates exemplary functionality carried out by the transaction server 468. The transaction server 468 maintains the dialer id, the last used value of the counter and the last access time in a table (see block 552). This table is used for protecting the network against replay attacks. This table is typically replicated across all transaction servers 468.

[00085] Upon receipt of the user credentials or authentication data from the netserver 470, in one embodiment of the invention, the transaction server 468 decrypts the password, and compares the received value of the counter with the value in stored in its database (see block 554). If the count sent by the dialer 466 is greater than the last count value stored in the database, then it is considered a genuine request (see block 556). If the count sent by the dialer 466 is equal to the last count value stored in the database, and the delta or time difference between the current system time and the time of last access stored in the database is less than a time window allowed, then again the request is considered genuine (see block 558). The transaction server 468 rejects the authorization request as a possible replay attack if the count sent by the dialer 466 fails these two conditions (see block 560). The transaction server 468 sends the authentication request along with the plain text password to the roam server 472 of Figure 9.

[00086] In the embodiment depicted in Figure 8 the transaction server 468 maintains a record of the customer's private key and, accordingly, decryption of the authentication data takes place at the transaction server 468, which may thus define a decryption server. However, certain customers may wish to not

provide their private key to any intermediaries such as the transaction servers 468. In these circumstances, the customer's private key is not provided to the transaction servers 468 but rather to the customer's roam server 472 that is typically at an in-house location. Accordingly, in addition or instead, decryption of the authentication data may thus take place in a similar fashion to that described above at the customer's roam server 472. An embodiment of a roam server 472 that includes encryption functionality is shown in Figure 9.

[00087] Referring in particular to Figure 9, reference numeral 570 generally indicates exemplary functionality carried out by the roam server 472. As the functionality substantially resembles the functionality 550 of Figure 8, like reference numerals have been used to indicate the same or similar features. When the transaction server 468 does not have access to the particular customer's private key, the transaction server 468 adds the necessary ECC attributes to the authentication request packet and sends it to the roam server 472 (see block 572). The roam server 472 decrypts the password and the checksum character using the ECC information and the private key stored locally (see block 552). The roam server 472 then performs the same functionality tests described above to determine if the count is valid (see blocks 554 - 560). The roam server 472 adds the decrypted count to the authentication reply packet (see block 574) so that the transaction server 468 can update its database with the latest value of the count (see block 576). Exemplary tables for implementing counter functionality are set out below.

[00088] A table dialer_counter_ts is typically used for replication. This table is created at each Transaction Server 468.

[00089] Table: dialer_counter_ts

FIELD NAME	DESCRIPTION
DIALER_COUNTER_TS_ID	A NUMERIC ID. REQUIRED FOR ORACLE SNAPSHOTS.
SERVER_ID	THE TRANSACTION SERVER ID. VARCHAR2(20).

DIALER_ID	THE DIALER ID IS OBTAINED FROM THE DIALER_ID SERVLET AT A WEB SERVER OF THE SYSTEM 454. VARCHAR2(10)
COUNTER	LAST USED VALUE OF THE COUNTER. VARCHAR2(5)
ACCESS_TIME	LAST ACCESS TIME

[00090] The last used value is typically stored in a database instance e.g., on "SESSION" machine. The SESSION machine is typically used to pull the entries from dialer_counter_ts tables in the transaction servers 468 and aggregate them into a single table. The SESSION machine also creates a unique snapshot corresponding to every dialer_counter_ts table in the transaction servers 468. These snapshots are typically named as dialer_counter_ts_<ServerId>, where ServerId is the id of the particular transaction server 468. The exemplary database instance SESSION is created with two identical machines on either coast to enhance fault tolerance.

TABLE: DIALER_COUNTER	
FIELD NAME	DESCRIPTION
DIALER_ID	THE DIALER ID IS OBTAINED FROM THE DIALER_ID SERVLET AT A SYSTEM WEB SERVER OF THE SYSTEM 454 AND IS USED FOR UNIQUELY IDENTIFYING THIS RECORD.
	VARCHAR2(10)
COUNTER	LAST USED VALUE OF THE COUNTER. VARCHAR2(5)
ACCESS_TIME	LAST ACCESS TIME

[00091] Each transaction server 468 typically replicates the dialer_counter table using Oracle snapshots. When a standard system is upgraded to

accommodate the present embodiment of the invention, the following exemplary modifications are typically made.

TABLE: SECURE_PAP	
FIELD NAME	DESCRIPTION
SPAP_ID	GENERATED ID THAT UNIQUELY IDENTIFIED THIS RECORD.
CUSTOMER_ID	CUSTOMER ID.
PUBLIC_KEY	PUBLIC KEY.
PRIVATE_KEY	PRIVATE KEY VALUE.
KEY_VERSION	KEY VERSION NUMBER.
ALGORITHM	ALGORITHM. FOR EXAMPLE, E AND A.
EXPIRATION_DATE	TIME/DATE WHEN THIS RECORD WILL EXPIRE. IF NULL, THIS RECORD WILL NEVER EXPIRE.
DESCRIPTION	DESCRIPTION ENTERED FROM DCT.
CREATION_DATE	TIME/DATE WHEN RECORD WAS CREATED.
MODIFY_BY	USER WHO MODIFIED RECORD.
MODIFY_TIME	TIME WHEN RECORD WAS MODIFIED.

TABLE: CUSTOMER	
FIELD NAME	DESCRIPTION
ENCRYPT_FLAG	0 = ENCRYPTION IS OPTIONAL, 1 = ENCRYPTION IS REQUIRED FOR THIS CUSTOMER

TABLE: DIALER_PROFILE	
FIELD NAME	DESCRIPTION
ENCRYPT_FLAG	0 = ENCRYPTION OFF, 1 = ENCRYPTION ON
SPAP_ID	REFERENCES SECURE_PAP TABLE

Encryption/Decryption functionality.

[00092] In the embodiment described above with reference to Figures 7 to 9, the dialer 466, transaction server 468, and roam server 472 include an ECC API that implements the ECC algorithm and provides an API for encrypting and decrypting passwords. Typically, the ECC implementation uses optimal normal basis mathematics for encryption/decryption. In certain embodiments, polynomial basis and optimal normal basis mathematics are combined to reduce the time for a mathematical inversion to the cost of a single multiply.

[00093] Referring in particular to Figure 10, reference numeral 580 generally indicates exemplary encryption functionality of the dialer 466. As shown at block 582, the encryption algorithm generates a random point on an ECC curve. This random point is then used for encoding the password and the checksum character (see block 584) to produce part of an ECC string <encoded password>. The dialer 466 encrypts the random point and transmits it to the netserver 470 (see blocks 586 and 587). Typically, a symbol transformation scheme is used for this encryption as described below.

[00094] In order to accommodate existing protocols, e.g., PPP, PAP, RADIUS, or the like, the password fields have printable US-ASCII characters. In certain embodiments, the characters are generated in such a fashion so as to conform to RFC 2486 standards. In these embodiments, when the password and checksum fields are encrypted, care is taken to generate the string with acceptable characters so that they may be applied in networks using standard protocols (see block 588). Accordingly, the following character transformation scheme may be used to perform this encoding. Each character to be encoded is first mapped into a value according to the table shown below.

#	SYMBOL	#	SYMBOL	#	SYMBOL	#	SYMBOL
0.	0	1.	1	2.	2	3.	3
4.	4	5.	5	6.	6	7.	7
8.	8	9.	9	10	A	11	B
12.	C	13.	D	14	E	15	F
16.	G	17.	H	18	I	19	J
20.	K	21.	L	22	M	23.	N
24.	O	25.	P	26	Q	27.	R
28.	S	29.	T	30	U	31.	V
32.	W	33.	X	34	Y	35.	Z
36.	A	37.	B	38	C	39.	D
40.	E	41.	F	42	G	43.	H
44.	I	45.	J	46	K	47.	L
48.	M	49.	N	50	O	51.	P
52.	Q	53.	R	54	S	55.	T
56.	U	57.	V	58	W	59.	X
60.	Y	61.	Z	62	~ (TILDE)	63.	` (GRAVE ACCENT)
64.	! (EXCLAMATION MARK)	65.	# (NUMBER SIGN)	66	$ (DOLLAR SIGN)	67.	% (PERCENT SIGN)
68.	^ (CARET)	69.	& (AMPERSAND)	70	* (STAR SIGN)	71.	((LEFT PARENTHESIS)
72.	) (RIGHT PARENTHE	73.	- (HYPHEN-MINUS)	74	_ (UNDERSCORE)	75	+ (PLUS SIGN)

#	SYMBOL	#	SYMBOL	#	SYMBOL	#	SYMBOL
	SIS)						
76.	= (EQUALS SIGN)	77.	{ (LEFT CURLY BRACKET)	78	[(LEFT SQUARE BRACKET)	79	} (RIGHT CURLY BRACKET)
80.	] (RIGHT SQUARE BRACKET)	81.	| (VERTICAL LINE)	82	\ (REVERSE SOLIDUS)	83	: (COLON)
84.	; (SEMICOLON)	85.	" (QUOTATION MARK)	86	' (APOSTROPHE)	87	< (LESS-THAN SIGN)
88.	, (COMMA)	89.	> (GREATER-THAN SIGN)	90	? (QUESTION MARK)	91	(SPACE)
92.	/ (SOLIDUS)	93.	. (FULL STOP)	94	@ (COMMERCIAL AT)		

[00095] The mapped value is then added to the corresponding byte in the random point and the modulus 95 is calculated (see block 590). This results in the character being mapped to another character in the above table. To decode the character at a decryption server, the corresponding byte in the random point is subtracted from the encoded character (see block 581 in Figure 11) and the modulus 95 of the result is calculated (see block 583). If the result is a negative number, then the value 95 is added to the result to obtain the original character (see block 585). By way of illustration, assuming "r" is the byte in the random point used for the encoding, and "x" is the original character, then,

```
Encode:  y = (x+r)%95
Decode:  x = (y-r)%95
         If (x < 0) then
           x = x+95;
```

[00096] The password field and the checksum character are encrypted with the random point during the encryption process at the dialer 466. Each one of these fields uses a different set of bytes in the random point for encoding. The password field uses the first set of bytes for it's encoding, and the checksum field uses byte 10 for it's encoding.

[00097] The checksum character is used for ascertaining the integrity of the dialer id and counter values. If the dialer id and the counter value are transmitted in the clear, a malicious person can alter these values and thereby defeat the protection against replay attacks. To address this problem, a checksum character is generated from the dialer id and counter value where after it is encoded using the random point (see block 592 in Figure 12). The encrypted checksum character is then transmitted as part of the user id string.

[00098] The checksum character is generated by the MD5 hash of the count value, the dialer id and the random point (see blocks 592 and 594 of Figure 12). Seven bits are then selected from the hash and then encoded with a single byte (byte #10) from the random point (see block 596) using the encoding methodology described above. The encoded bits are then dispersed among the last seven bytes of the encypted point (see block 598) and transmitted as part of the user string (see block 599). When the dialer 466 sends the encoded data to the transaction server 468 or roam server 472, as the case may be, they validate the dialer id and counter value by independently generating the checksum (see block 588 in Figures 8 and 9) and compare it with the checksum sent by the dialer 466 (see block590) and reject if they don't match.

[00099] Returning to the dialer 466 and to Figure 10, the encoded strings are then concatenated as follows to create an ECC string:

[000100] <encoded password><encrypted and encoded x coordinate of the random point with encoded checksum bits in the last seven bytes>

[000101] Thereafter, the dialer 466 concatenates the ECC string with the dialer id and the counter value and transmits it in the userid and password fields of the protocol, e.g. PAP. For example, <encoded password><encrypted and encoded x coordinate of the random point with encoded checksum bits in the last seven bytes><dialer id><counter value> .

[000102] It will be noted that the methodology set out in Figure 10 produces an encrypted string that is of such a string length, and includes characters of such a nature, that the encrypted string may be communicated using LEGACY systems

[000103] The encryption logic is typically encapsulated in an ip_spap_encrypt()method with the following signature:

char * ip_spap_encrypt(const char *algorithm, const char public_key, const char password, const char *dialer_id, const char *counter, char **plain_point, char **encrypted_point, int *returnCode);

where

algorithm is the algorithm to be used. "S" for Secure PAP

public_key is the ECC public key (from config.ini)

password is the plain-text password

dialer_id is the id of the dialer (obtained from the dialer id servlet)

counter is the count of dial attempts (incremented by the dialer for each dial attempt)

plain_point - If this field is left empty, a new random point is generated. This field points to the random point used for the encoding on return.

encrypted_point - If this field is left empty, the plain point and the public key is used to generate the encrypted point. This field points to the encrypted point used by the method on return.

returnCode 0 if the call is successful, a non-zero code is provided. The method returns the ECC string is returned when successful and a null otherwise.

[000104] The decryption logic is encapsulated in the ip_ spap_decrypt()method. The method have the following signature:

char * ip_spap_decrypt(const char *algorithm, const char private_key, const char ecc_string, const char *dialer_id, const char *counter, int *returnCode);

where

algorithm is the algorithm to be used. "S" for secure pap

private_key is the ECC private key (from securepap table or esp_key_pair.txt file)

ecc_string is the string returned by the encrypt() method

dialer_id is the id of the dialer (obtained from the dialer id servlet)

counter is the count of dial attempts (incremented by the dialer for each dial attempt)

returnCode 0 if the call was successful; non-zero code otherwise

The method returns the plain text password when successful and a null otherwise.

Dialer Customization Form

[000105] As mentioned above, the customers 456 use a web form for requesting a customized dialer configured to communicate using Secure PAP. This web form typically contains fields that can be used for specifying the required customizations. The web form may include the following exemplary fields:

Enable Secure PAP encryption: (Y/N)

Public Key:

Key Id: (0-9)

Dialer Customization Tool

[000106] During the customization process, an administrator of the access broker system 454 has the option of generating a dialer 466 that will use Secure PAP. If enabled, the following exemplary fields may be set in a config.ini that is typically packaged with the dialer 466:

[processing facility identification e.g., iPass]

EncryptFlag=yes

Algorithm=S

KeyVersion=0

PublicKey=BwAAAMGdqYx2lxhWtEQMdDHhvwU=&AQAAAFdd40uLQMD1UTtyBqDHY=

[000107] These values are also stored in the transaction server database so that the transaction server 468 can decrypt the password sent from the corresponding dialer 466 of a particular customer 456. In the present embodiment, only the public key is stored in this file, as the private key is kept secret for the encryption to be secure.

[000108] In addition to enabling Secure PAP, the customization tool also provides the option of setting the algorithm used and the key version. For example, the following encryption algorithms may be supported:

A for no encryption.

E for Elliptic Curve Encryption

S for ECC compatible with Unique Session ID

U for Unique Session ID

[000109] In practice, A is primarily for testing and debugging purposes. E is used for encrypting the password when the dialer does not have the dialer id. U is not an encryption algorithm, but is used to identify the unique session id, as discussed in more detail below. The key version starts at zero, but is incremented every time a new key-pair is desired for an existing dialer profile. The dialer 466 stores the ECC keys and other information in a secure_pap table. This table is then replicated to the transaction server 468 via Oracle snapshots. A new key-pair is generated if the private key has been compromised. When the security of the private key is compromised, the dialer support team should take the following actions:

1. Set an appropriate expiry date for the compromised key. This should be sufficient to ensure all dialers 466 using the compromised key can still use the key one last time. The dialers 466 connect to the Internet using the old key, and retrieve the config.ini file with the new key from the update server.

If the customer 456 is using the roam server 472 to decrypt the password, the customer 456 typically manually removes the compromised key from the esp_key_pair.txt file after the expiry date.

2. Generate a new key pair or ask the customer 456 to generate a new key pair and send the public key to the access broker system 454.

3. Use the DCT tool and replace the public key (use a new key id). Build the dialer.

Dialer

[000110] The dialer 466 checks the config.ini file to determine whether or not it should be encrypting passwords. If Secure PAP is enabled, then the dialer 466 encrypts the password using the public key from the config.ini file and by invoking the ip_ spap_encrypt() method. The method creates the ECC string and returns it. The dialer 466 concatenates the ECC string with the dialer id and the counter value. The first sixteen characters of the ECC string are placed in the password field and the rest of the string is placed in the prefix field (with 0S or 0E prefix). The dialer 466 uses algorithm "E" until it obtains a dialer id. The prefix is included after all system and routing prefixes, but before the customer prefixes. The dialer 466 does not encrypt the password and does not create the Secure PAP prefix if the POP being dialed has a prefix that is not compatible with and PAP prefix in the phonebook. A sample username, which includes the encryption prefix is a follows:

UserID: IPASS/0S Axrt50zTxca546hjdgkbxcjc^_d0we/joe@ipass.com

Password: x35~!4Qu{xy71]D8

where KeyVersion=0 and Algorithm=S.

[000111] If the access broker system 454 determines that no encryption is needed, it creates a unique session id from the dialer count and places it in the prefix field. A sample username, which includes the unique session id prefix is as follows:

UserID: IPASS/0UAxrt5AB2/joe@ipass.com

Password: thisisabigsecret

where KeyVersion=0 and Algorithm=U.

The dialer 466 stores the plain_point and the encrypted point in its local storage.

[000112] When a redial is attempted, the dialer 466 increments the counter and invokes the ip_spsp_encrp() method using the plain point, and encrypted point.

Customer Resolution

[000113] The customer resolution process checks for a prefix of the form [0-9][A-Z]*/ If no such prefix is found, and the customer 456 does not require password decryption, the customer resolution operates as normal. If the prefix is found, the last 8 bytes up to the first slash (/) are stripped out and stored as the unique session id field. The customer resolution code may create the unique session id field with the following contents: 0S<dialer_id><counter>. The integer is stripped and stored as key identifier field. The algorithm is stripped and stored as a separate field.

Dialer Counter Replication

[000114] Secure PAP embodiment depicted in the drawings uses the dialer_counter table for protection against replay attacks. Each transaction server database contains a dialer_counter_ts table. The transaction server 468 inserts a new row into this table whenever it receives a successful authentication request with a Secure PAP prefix. The contents of this row include the server_id, the dialer_id, the counter and the system time (in GMT).

[000115] The SESSION database contains a snapshot for the dialer_counter_ts table at every transaction server 468. These snapshots are typically named: dialer_counter_ts_<SERVER_ID>, where <SERVER_ID> is the id of the particular transaction server 468.

[000116] A "refresh" tool is provided for refreshing the snapshots from the transaction servers 468. The dialer_counter_ts_<SERVER_ID> would have "ON INSERT" PL/SQL trigger that would update/insert the dialer_id, counter, and access_time from the inserted row into the dialer_counter table if the value of the counter being inserted is equal to or greater than the value of the counter in the dialer_counter table. The transaction servers 468 use the refresh tool to refresh the dialer_counter snapshot from the SESSION database. The

dialer_couter table is then cached by the transaction servers 468 for faster access. Any changes to records in dialer_counter table at runtime take immediate effect. This is accomplished using the same mechanism used in other components of the access broker system 454 using database triggers and the cache_update table.

Transaction Server

[000117] On startup, the access broker system 454 reads all private keys from the database into a local cache for efficient lookup. It also has an additional attribute in the customer cache to indicate if a certain customer 456 requires password encryption or not. The transaction server 468 also caches the dialer_counter table. Any changes to records in these tables at runtime take immediate effect. This is accomplished using the same mechanism used in other components of the access broker system 454 using database triggers and the cache_update table.

[000118] If the encrypted prefix field specifies the 'S' algorithm, the transaction server 468 concatenates the contents of the password field to the encrypted prefix field constructed by the customer resolution process and creates the "ECC field". The ECC field contains

<encoded password><encrypted and encoded x coordinate of the random point><encoded checksum character>

[000119] The transaction server 468 locates the private key for the appropriate customer 456 using the key index. If the private key is found in the database, it calls the ip_spap_decrypt() method to decrypt and decode the password. The password field is then overwritten with the plain-text password before it is sent to the roam server 472.

[000120] If the private key is not located in the cache, the transaction server 468 typically adds the following fields to the authentication request packet and sends it to the roam server 472: algorithm, key index, the ECC field (as password), dialer id, counter, value and access time of the counter last used (from the database), and the "decrypt_at_roamServer" flag set to "yes".

[000121] The transaction server 468 then stores the authentication details in the ip_auth_trans table and the dialer_counter details in the dialer_counter_ts table. The Transaction server 468 typically inserts a new dialer_counter_ts record every time as inserts are usually faster than updates.

[000122] When the transaction server 468 receives the account request, it uses the customer resolution process to create the unique session id and adds it to the packet as "ipass_session_id". The tr_userid field contains the raw_userid.

ESP Tool

[000123] The ESP command line tools are used by the customers 456 in conjunction with their roam servers 472, the DCT team, and the QA team to generate public/private key pairs and test the encryption and decryption algorithms.

esp_genkey (for customers 456 with roam servers 472):

[000124] This tool prints the public/private ESP key pair to a file named *esp_key_pair.txt*. This file resides in the */usr/ipass/keys* directory on Unix, and in the *IPASS_HOME/keys* directory for Windows. The keys must also be submitted to the access broker system 454 via, for example, a secure website so that the dialer 466 can be built with the public key. Typically, a secure backup of the private key is also maintained.

esp_genkey_dct:

[000125] This tool prints the public/private ESP key pair to standard output. It is printed in a format that meet the requirements of the DCT. An example output is:

1

Public Key:BgAYVK1azUt8comk41GzLw=&ADIkGfMgNChM4vY6+nLgTqo=

Private Key:AQAAAAZOSNH13PaG3NuqGbU7TY0=

[000126] The first line contains a "1", indicating success in key generation. When an error occurs, that output is then of value "0".

esp_qa:

[000127] This tool has several command line options available for testing the ECC API. An example sample of the option supported:

esp_qa genkey

esp_qa encrypt [-a <algorithm> -d <dialer_id> -c <counter>] -k <public_key> -t <text>

esp_qa decrypt [-a <algorithm> -d <dialer_id> -c <counter>] -k <private_key> -t <text>

esp_qa testipg [-a <algorithm> -d <dialer_id> -c <counter>] -k <public_key> -t <text> -u <uid

@domain>

esp_qa test -t <text>

Option in brackets[] are optional. Each esp_qa command are described as follows:

genkey - Generate a public/private key pair.

encrypt - Encrypt text (password) with the given public_key.

decrypt - Decrypt text (password) with the given private key.

testipg - Executes the "Encrypt" then runs the check-ipen command for the given user.

test - Basic ECC API test. Runs the genkey, encrypt, and decrypt for algorithm S.

Roam server

[000128] The roam server 472 typically checks for the presence of the "decrypt_at_roamserver" field in the packet received from the transaction server 468. If the field is present, the roam server uses the "key index" field from the packet and fetches the private key from the esp_key_pair.txt file. The ECC string along with the private key, dialer id and counter value is passed to ip_spap_decrypt() method. The ip_spap_decrypt() method decodes and decrypts the password. The plain text password is then used by the roam server 472 to authenticate the user.

[000129] Returning to Figure 6, once the dialer 466 has performed the methodology set out above, the authentication data is communicated to the NAS 532 where after it is sent to an authentication server 600 of the remote ISP 506. In the normal course of operations, the NAS 532 at the remote ISP 506 would reject the supplied authentication information. However, as illustrated in Figure 6, the netserver 470 intercepts the authentication information to facilitate recognition of this authentication information as a roaming user authentication request and not a regular user request.

[000130] The authentication server 600, in conjunction with the netserver 470, parses the received authentication information to determine a roaming domain name or routing prefix associated with the roaming user 502. Should such a domain name or prefix be present, the user's authentication information is encrypted as set out above, and sent from the netserver 470 to the transaction server 468 via a secure socket layer (SSL).

[000131] The transaction server 468 may use a customer routing prefix in the session identification to route the request. Instead, the transaction server 468 may perform an Internet Protocol (IP) look-up and routes the authentication request to an appropriate home ISP 504. More specifically, the transaction server 468 receives the encrypted authentication request from the netserver 470 at the remote ISP 502, and decrypts this request as described above with reference to Figures 7 to 9. The transaction server 468 then determines the "home" ISP 504 by matching the roaming domain name or routing prefix of the desired home ISP 504 against a current list of participant domain names and IP addresses. If the match is successful, the authentication request is encrypted and sent via SSL to the roam server 472 that resides at the home ISP 504. In the event that the identified roam server 472 does not respond within a specific period, the transaction server 468 will attempt to contact an alternative roam server 472 at the ISP of the relevant domain.

[000132] The roam server 472 at the home ISP 504 then decrypts the authentication request sent from the transaction server 468, as described above, and submits the authentication request to the home ISP's regular authentication

server 602 as if it were a terminal server or NAS 532 owned by the home ISP 504 using a customer prefix. The authentication server 602 of the home ISP 504 responds to the request by providing an "access permitted" or an "access denied" response based on the validity of the user name and password included within the authentication request (see Figure 8). The response from the home ISP's authentication server 602 is received by the roam server 472, encrypted, and sent back to the transaction server 468.

Unique session identification

[000133] An exemplary method and system to associate a plurality of transaction data records is described below. The method and system describes the generation and use of a unique session id which is typically used in combination with the encryption/decryption methodology described above.

[000134] As mentioned above, communication protocols such as, for example, Point-to-Point protocol (PPP), Password Authentication Protocol (PAP), Challenge-Handshake Authentication Protocol (CHAP), Remote Authentication Dial In User Service (RADIUS) protocol, Terminal Access Controller Access Control System (TACACS) protocol, Lightweight Directory Access Protocol (LDAP), NT Domain authentication protocol, Unix password authentication protocol, HyperText Transfer Protocol (HTTP), HyperText Transfer Protocol over Secure sockets layer (HTTPS), Extended Authentication Protocol (EAP), Transport Layer Security (TLS) protocol, Token Ring protocol and Secure Remote Password protocol (SRP) make provision for a user identification string. Although the size or length of characters that each different protocol allows may vary, the lowest common denominator in size supported by the exemplary protocols listed above is typically about 63 characters. In these circumstances, provision of a unique user session identification would enhance authentication, accounting and SQM processing.

[000135] In the application of above protocols to the exemplary multi-party service access environment 450, the user identification string is included, and is thus common, in all relevant transactions data records generated by the various participants, such as the transaction servers 468, the service providers 452 and

the customers 456. However, in certain circumstances, although the prior art user identification string used in these protocols may be uniquely associated with a particular user of multi-party service access environment 450, it is not uniquely associated with a particular single user session. For example, due to network timeouts and packet retry algorithms, it is often the case that a single transaction data record is sent to a transaction servers 468 several times and, if any one or more of these records is defective, multiple instances of a record relating to the same single user session may exist at the settlement system 474. Further, in an attempt to re-send a perceived failed communication attempt, certain NASs 470 (see Figure 6) actually change the user session identification string thereby resulting in different transaction data records for the same single user session. The aforementioned are merely two examples of unsatisfactory accounting records but it will be appreciated that there may be a host of other circumstances.

[000136] In accordance with another embodiment of the present invention, relevant transaction data records generated in response to a single user session include a common unique session identification. In certain circumstances, this session identification may provide strong, but not necessarily absolute identification of an individual user's usage information and the unique user session identification should at least be unique within certain parameters. For example, the unique user identification my be unique for a given time period so that all records generated during that time period may be associated and processed using the unique user session identification.

[000137] Typically, for the exemplary protocols mentioned above, the user identification string includes, not only a user name and password of the user accessing the network, but also routing information including the customer realm. The user ID or identification string used in the exemplary multi-party service access environment 450 is typically as follows:

<FacilityRoutingPrefix>/[<FacilityLocationPrefix>]/[<CustomerRoutingPrefix>]/[CustomerPrefix(s)]/<EndUserName>@[<NonRoutingCustomerDomain>] |[<CustomerRoutingDomain>]

Wherein,

<FaclityRoutingPrefix> is a proprietary prefix that is used by the ISPs 458, wireless access providers 460, content distribution provider 462, E-Commerce provider 464, or the like (the access providers) to route traffic to the network of the access broker system or facility 454.

<FacilityLocationPrefix> is a prefix used by the facility to determine the location of points or nodes providing access to the facility 454.

<CustomerRoutingPrefix> is a prefix used by the access or service providers 452 to route traffic to the customer site.

<CustomerPrefix(s)> is a/are prefix(es) used by the customer 456 for their internal routing.

<EndUserName> is the login user name of the end user 502 using the facility 454.

<CustomerRoutingDomain> is a domain used by the system 454 to route traffic to the customer site. The user ID string includes either the <CustomerRoutingPrefix> or the <CustomerRoutingDomain>.

<NonRoutingCustomerDomain> is a domain used by the customer 456 for their internal routing.

[000138] An example of one of the possible ways of fitting the unique session identification in the user identification field of one of the above protocols is now described. It will however be appreciated that the inclusion of the unique session identification may be implemented in other ways. An example of an alternative solution is implemented when the dialer uses the E type algorithm for password encryption. The E type algorithm includes the encrypted random point in the username. The encrypted random point provides strong, but not necessarily absolute identification of the individual users session, and so is used as the unique session id.

[000139] As mentioned above, the lowest common denominator available string length for proprietary information supported by the exemplary protocols is typically about sixty-three characters. The unique session id should fit within the limits imposed by the username field.

[000140] In order to generate the unique session identification (see block 802 in Figure 20), the connection application 466, in the exemplary form of a connect dialer, resident at each service provider 452, obtains a dialer identification which identifies the connect application 466 from a servlet in the web server 806 of the access broker system 454 (see Figure 15). The dialer identification is typically also a unique dialer identification. The dialer identification is stored in a user preference file and, when the dialer is initially installed; the dialer identification in the user preference file is typically empty. The first time the dialer 466 connects, for example, to the Internet, it typically requests a new dialer identification from the web server 806 (see block 800). In the embodiment shown in the drawings, the dialer does not create a unique session identification until it obtains the unique dialer identification from the web server 806. Accordingly, in this embodiment where the dialer identification forms part of the unique session identification, the first successful session from the dialer 466 would not contain a unique session identification. The dialer 466 would however have its dialer identification for any subsequent attempts.

[000141] In addition to its own dialer identification, the dialer 466 also includes a counter 467 that is internally maintained and stored in the user preference file. The counter 467 is incremented for each dial attempt (see block 802). The dialer 466 using its dialer identification and the counter generates a session identification indicator, defined by eleven characters (see Figure 18) in the present embodiment, at each subsequent dial attempt. As the counter 467 is incremented at each dial attempt, the dialer generates a globally unique session identification: <dialer id><counter>(see block 802). In this embodiment, the session identification is prefixed by an identifier, e.g., three characters such as "OU" associated with the facility or access broker system 454, which are stripped off by the transaction server 468 before the user identification string is passed onto the roaming server 472 (see Figure 5). Thus, when the unique session identification includes eleven characters, eight characters would be available for the dialer identification and counter.

[000142] Both the exemplary dialer identification and counter use numbers with radix 64. The symbols used for this numbering scheme include A-Z, a-z, 0-9, & and ^. The counter 467 is incremented prior to each dial attempt and the dialer identification is pre-filled with zeroes and, in the present embodiment, defined by a five digit entry. Accordingly, three digits remain for the counter 467. Accordingly, the five digits used for the dialer identification would enable 1073741824 unique dialer installs (more than a billion) and the three digit counter enables 262144 dial attempts (the counter would reset after 23 years, assuming 20 attempts a day). During this period, the session identification would thus uniquely define each user session. It is however to be appreciated that the number of characters allocated or used for the unique session identification may vary from system to system dependent upon the type or types of protocols that the system accommodates.

Transaction Record Processing

[000143] Figure 14 is a block diagram illustrating the accounting and settlement procedures, according to an exemplary embodiment of the present invention, which may be facilitated by the access broker system 454.

[000144] When a roaming user 502 connects to the remote ISP 506, the terminal server (or NAS) 470 managing the session generates a transaction data record that includes the user identification string, and thus the eleven character unique session identification, and sends this information to the authorization server 600. The authorization server 600, in conjunction with the netserver 470, parses the accounting information to determine a roaming domain name and prefix associated with the roaming user. Should such a domain name or prefix be present, the user's accounting information is encrypted using an algorithm from RSA Data Securities, and sent from the netserver 470 to a transaction server 468 via secure socket layer (SSL).

[000145] When a roaming user 502 disconnects from remote ISP 506, the terminal server (or NAS) 470 managing the session generates a transaction data record that includes the user identification string, and thus the eleven character unique session identification, and sends this information to the authorization

server 600. The authorization server 600, in conjunction with the netserver 470, parses the accounting information to determine a roaming domain name and prefix associated with the roaming user. Should such a domain name or prefix be present, the user's accounting information is encrypted using an algorithm from RSA Data Securities, and sent from the netserver 470 to a transaction server 468 via secure socket layer (SSL).

[000146] A transaction data or accounting record is then communicated, in near real-time, to the transaction server 468 utilizing SSL, where the accounting records are stored in the database. All the various components or participants in the multi-party service access environment 450 receive the user identification string, and thus the unique session identification, which then accompanies the transaction data record associated with the single user session when the transaction data record is sent to the settlement system 476. Thus, transaction data records sent from various different participants include an identifier that identifies the single user session from which they arise.

[000147] These accounting records are further processed by the settlement system 476 to produce Call Detail Records (CDRs). Each call detail record provides detailed usage reporting regarding the identity of the roaming user 502, when the relevant service access occurred, the location of the service access, the length and cost of each service access session, and the time of the service access (e.g., local or GMT time).

[000148] Multiple transaction servers 468 provide accounting or transaction data records to the settlement system 476, which utilizes these records to generate bills (or invoices) to customers 456, and also to make payments to service providers 504. It is, however, to be appreciated that accounting information sent to the transaction server 468 may, for various reasons, be incomplete, differ from one ISP to the next, be sent more than once and so on. Thus, a variety of different, and possibly incomplete, records relating to the same single user session may be received by the transaction server 468.

[000149] Naturally, identifying or associating all transaction data records arising from a particular user session is advantageous in that the settlement

system 474 generates bills and distributes them among customers 456 so that they can make payments to the settlement system 474, and in turn bill their customers if appropriate. Similarly, the settlement system 474 makes payments to the remote (or visitor) ISPs or other service providers 452 for accrued access time used by roaming users. The settlement system 474 may further guarantee payment for authorized use by a roaming user. An operator of the settlement system 476 thus acts as a secure, trusted entity providing a mechanism for facilitating financial settlement of service access transactions between multiple parties. The settlement system 476 implements numerous automatic functions and operations so as to enable the settlement in a timely, automated and convenient manner. Further details regarding the operation of the settlement system to facilitate such settlement or service access transactions will be described in detail below.

Physical Architecture

[000150] Figure 15 is a diagrammatic representation of the physical architecture of the access broker system 454, according to an exemplary embodiment of the present invention. Multiple transaction servers 468 are shown to reside on one or more server machines 810, each of which has access to an associated database 812. A web server and phonebook server reside on the server machine 806, and are accessible by remote internal users 814 and the customers 456. The web server operates to generate and deliver web pages (e.g., HTML documents) to both the remote internal users 814 and the customers 456. As described above, in one embodiment of the invention, a servlet on the web server residing on machine 806 provides a unique connection application identification, in the exemplary form of a dialer identification, to each dialer or connection application 466 residing with the services providers 452. The phonebook server (part of the phonebook management system 480) operates to maintain and update the electronic phonebooks of customers 452, and accordingly both receives and publishes updates to and from service providers 452, and publishes such updates to customers 456.

[000151] The settlement system 476, and a collection of internal users 816 are shown to reside behind a firewall 818. Specifically, the settlement system 474 is hosted on one or more server machines 820 that have access to a central database 822.

Overview - Settlement System

[000152] Figure 16 is a block diagram illustrating the architecture of a settlement system 474, according to an exemplary embodiment of the present invention. The settlement system 474 comprises a back-end applications 824, front-end applications 826, data aggregation and reporting applications 828 and system interfaces 830.

[000153] The back-end (or server-side) applications 824 are shown to include a settlement application 832 that determines a transaction price, updates account balances for all parties involved in a transaction, and verifies credit limits, a billing application 834 that closes an accounting cycle, applies periodical fees, generates billing reports, including invoices and call detail records (CDRs), and publishes billing reports to the web, and an auditing application 836 that verifies business rules and structural integrity of the central database 822. The settlement application 832 is shown to embody the flexible pricing engine 476.

[000154] In the present embodiment, the settlement application 832 is responsible for normalization, summarization and verification functions. The normalization function includes converting accounting data received from multiple transaction servers 468 into a single format CDR to be used for billing, identifying parties involved in a service access transaction, and defining the price that the access broker system 454 owes to a provider 452 and the price that a customer 456 owes to the access broker system 454 for a particular service access transaction. The summarization function involves applying buy and sell prices to account balances for all parties involved in a service access transaction, and updating appropriate account balances. The verification function includes the verification of credit limits.

[000155] The settlement system 474 operates to provide near real-time settlement of service access transactions to allow for the near real-time revenue and account tracking by both providers 452 and customers 456.

[000156] In certain embodiments of the invention, the settlement system 474 includes the flexible pricing engine 476 that supports a flexible pricing model, which has the following features:

1. A variety of data structures dependent on, for example, the customer 456, the service provider 452, the location of the service access, the type of service access (e.g., dialup modem, ISDN, DSL), or usage accumulated during a particular cycle for a particular customer 456.

2. Any combination of (a) usage (e.g., a function of rate and session length); (b) transactional (per transaction); and (c) subscription-based or flat pricing (e.g., one price for all usage during a billing cycle for a customer 456 or one or more prices per each user for a customer during a billing cycle).

3. Offered discounts and promotions.

4. A variety of fees, such as start-up fees, monthly fees and minimum monthly commitments.

5. Multi-tiered pricing schemes, or intra-provider roaming, where buy and sell rates for a particular location depend on the provider 452 and whether the service user/customer 456 of the service access belongs to a further customer 456, its affiliate, or their customer.

[000157] The flexible pricing engine 476 is database-driven, thus allowing implementation of new pricing models by loading the appropriate plan into pricing tables (not shown) maintained within the central database 822. More specifically, the flexible pricing engine 476 facilitates a multi-tiered pricing model, whereby rates for a single service access transaction may be applied across multiple tiers of consumer (or customer) according to multiple criteria. These criteria may include, *inter alia*, any combination of usage (e.g., accumulated usage time or value total) pricing and transactional (e.g., an accumulated total number of transactions) pricing.

[000158] Returning now to Figure 16 and the front-end applications 826, a data management application 838 is utilized by various functional units of the access broker to perform business processes and to view data for information purposes. To this end, data management application 838 may provide various user interfaces to manage information related to customers 456 and access points, and to perform accounting and administrative functions.

[000159] An order processing application 840 provides user interfaces to customers 456 (e.g., solution partners 488 or resellers) to place orders for new corporate customers.

[000160] The data aggregation and reporting applications 828 include several processes that summarize data on a daily or monthly basis to enable operational, functional and network load reporting.

[000161] The system interfaces 830 have a loader application that includes a transaction server loader 842, a provider loader 844 and accounting system interfaces (not shown). Dealing first with the transaction server loader 842, a "data loader" component pulls accounting records in the form of transaction data records, including the unique session identification, from the databases 812 of the respective transaction servers 468 to the central database 822 for processing. Multiple transaction server or batch loaders 842 may be implemented as distributed database links, and the accounting or transaction data records are pulled via the loaders 842 in near real-time.

Overview - Data Model

[000162] Figure 17A is a block diagram illustrating an exemplary data model 845 including customer tables 846, access point tables 848, pricing tables 850, CDR tables 852, accounting tables 854, authentication transaction storage area or tables 856, batch history storage area or tables 858, and SQM storage area or tables 860.

[000163] The network components in the access broker system 454 may, in certain embodiments, strip the routing prefixes from the transaction data records. Some of these components may also truncate the user identification string. The Unique session id prefix is neither a routing prefix nor at the end of

the username, hence it is neither stripped nor truncated The user identification string is thus processed to remove these defects before it is used to uniquely define the user session. A modified user session identification is constructed using as many of the following components that are available:

<AuthCustomerId>/<UniqueID>/[CustomerPrefix(s)]/<EndUserName>@<NonRoutingCustomer Domain>

Wherein,

<AuthCustomerId> is the authenticating customer identification, produced by the customer resolution process.

<UniqueID> is the unique session identification code, 0Uxxxxxxx/, prefix generated by the connection application 466 as described above.

<CustomerPrefixe(s)> are prefixes used by the customer for their internal routing as described above.

<End User Name> is the user identification of the end user connecting to the access broker system 454 as described above.

<NonRoutingCustomerDomain> is the domain used by the customer for internal routing, as described above.

[000164] Referring in particular to Figure 17B, provider loader 844 receives call detail records (CDRs) or transaction data records, including the unique session identification, from the providers 452 in a batch form. This CDR data is pre-processed by the provider loader 844, which may retrieve the data from an appropriate FTP site and convert it into the same format as the data received from the transaction servers 468. In particular, the transaction server 468 constructs, each time a user access session is authorized as described above, a modified user session identification and stores it a session_id field in authentication transactions tables 856 and a session_id field in account transaction tables 854 (see Figure 17A). It will be appreciated that, for each modified session identification stored in the authentication transactions table 856, corresponding transaction data records should be received by the settlement system 474 for processing. In a similar fashion to the transaction server 468, the batch loaders 842, 844 respectively construct or build a modified

transaction data record from each transaction data record received from the transaction servers 468 of the service providers 452 (see Figure 5 and 17B). The modified session identification from the loaders 842, 844 are stored it in a session_id field in the batch history tables. Likewise, the SQM process constructs the modified session identification and store it in a session_id field in an SQM table 860.

[000165] The use of the unique session identification including its unique code may be used in addressing the following issues.

Missing Accounting records

[000166] Missing accounting/transaction data records (see block 862 in Figure 19) may arise for various reasons such as delivery failure, malformed records, misrouted records, or the like. Delivery failure may occur when the Internet connectivity from the ISPs (e.g., the netserver 470 in Figure 6) is disrupted, thereby blocking the delivery of the transaction data record to the settlement system 476. A connectivity outage that persists for more than a few minutes typically causes the netserver 476 to discard the transaction data record due to minimal transmission retry capabilities. When using the RADIUS protocol, malformed records are typically discarded at any of several intermediate points including the authentication server of the service provider (e.g., the authentication/authorization server 602 or netserver 470 (see Figure 6)). Misrouted records are records not sent due to an improper configuration of the ISPs authentication server 602 either accidentally or with fraudulent intent.

[000167] As every access session by the user first requires authorization, each session_id field in the authentication transaction tables 856 should include a corresponding session_id field in the acct_trans table. Accordingly, by associating, matching, correlating, investigating, the session_id fields, missing accounting/transaction data records can be determined. In the embodiment depicted in the drawings, missing accounting/transaction data records typically would have an authentication request record in the authentication transaction or auth_trans table 856, but no matching accounting start and/or accounting stop records in the accounting or acct_trans table 854. Thus, by

searching for all session_id fields in the acct_trans table that correspond to each session_id field in the auth_trans table, missing accounting/transaction data records may be found (see blocks 864-872).

Inappropriate accounting records

[000168] Inappropriate accounting/transaction data records may be received by the settlement system 474 (see Figure 6) usually due to inappropriate configuration of a provider's authentication server (AAA server), e.g. server 600 in Figure 6. An inappropriate configuration typically causes the provider's authentication server 600 to send all accounting/transaction data records to all proxies instead of just the one responsible for the authentication of the user access session. In these circumstances, no session_id field is present in the auth_trans table of an incorrect recipient. As these accounting/transaction data records typically do not have a corresponding authentication record, they may be identified with relative ease and, for example, a customer support team can resolve the configuration problem with the provider and prevent recurrence of such incorrect transmissions. In these circumstances, the methodology shown in Figure 19 may be used except that the auth_trans table is searched for a unique session identification corresponding to an entry in the acct_trans table.

Duplicate Accounting records

[000169] Duplicate accounting records are multiple transaction data records that describe the same single user access session. In the embodiment depicted in the drawings, duplicate transaction data records are actively filtered by the settlement system 474 using a relatively simple algorithm that matches six "key" fields of each real-time accounting/transaction data record against all other real-time accounting/transaction data records that have been received within the previous 30 days. The exemplary fields used are: RADIUS Session-Id, Provider ID, NAS IP Address, User, Domain (user auth realm), and Session Time.

[000170] In certain embodiments, when all six fields match those in an already-rated record in the CDR table, the current record is marked as a duplicate and discarded.

[000171] Duplicate accounting records may arise for a variety of reasons:

[000172] Accounting/transaction data records must be acknowledged through the timely transmission of an Accounting-Response message to the sender. Unfortunately "timely" is not defined by the RADIUS specification and different vendors and configurations may resend unacknowledged accounting transactions a few seconds, hours or even days later. When the accounting request was actually received by the settlement system but the acknowledgement was lost or malformed, the originator may resend multiple copies of the accounting record. All such records are captured by the receiving transaction server 468 and eventually retrieved by the settlement system 474 for processing. Peculiar variations on this class may occur which elude the settlement duplicate filtering algorithm wherein the sending NAS 532 sends an updated (e.g., incrementally longer) session time with each retransmission or the RADIUS Session-Id changes between retransmissions. In some cases the NAS 532 does not send a consistent NAS IP address and, in these circumstances, another attribute (e.g., Called Station Id or Provider Id) is used to associate the access session with the service provider or ISP. Such cases reduce the usefulness of the NAS IP for duplicate detection.

[000173] Duplicate accounting records may be sent by the "batch" providers whose accounting feeds are assumed to be duplicate-free. Duplicate accounting records may be manually injected into the settlement system 474 when batches of records are sent by real-time accounting providers to complete their accounting responsibility when they have failed to deliver accounting records for one of the reasons described above. In these cases, arbitrary datasets may be sent by the service providers 452, which must be specially processed by personnel at the access broker system 454 to prepare them for submission in a data normalization process. Such datasets may contain data describing both previously reported sessions as well as the missing sessions for which the correction is attempted. Because these record batches are typically preprocessed as one-offs, little control exists to prevent duplicate injection. It

will be appreciated that this process can be automated in view of the unique session identification.

[000174] Some service providers may admit duplicate transaction data records to the access broker system 454 due to irregular use of key duplicate fields. For example, in certain circumstances, service providers fill the NAS IP attribute with random data that thus adversely influences the duplicate filter criteria. Other anomalies such as inconsistent session id generation or a failure to fix session duration at the time of user disconnect may generate duplicates that appear to correspond to distinct sessions. Once again, the unique session identification can assist in resolving these problems.

[000175] As shown by way of example in this document, duplicate accounting/transaction records are actively filtered by the settlement system 474 using an algorithm that matches six "key" fields of each real-time accounting/ transaction data record against all other real-time accounting/transaction data records that are received within the previous 30 days. Using the unique session_id field that uniquely identifies each approved single session, enhanced accuracy may be obtained.

Duplicate Alias Records

[000176] Duplicate alias records arise when an algorithm to detect duplicates inappropriately identifies a record as duplicate. For example, such cases can arise when a service provider's NAS (for example the NAS 532 of the ISP 510 in Figure 6) does not generate or reuses session identification data values within a short time. In addition to, or instead of, any session identification generated by the service provider that is not reliable, the unique session identification of the embodiment of the present invention, which uniquely defines each single access session, may be used by the duplicate detection algorithm to reduce the occurrence of duplicate alias records. In particular, the modified unique session identification in the session_id tables may be used to at least substantially reduce duplicate alias records as each session is uniquely identified.

Invalid Session-Length Records

[000177] It will be appreciated that all accounting/transaction data records received by the settlement system 474 relating to the duration of an access session may not always be complete. For example, an accounting/transaction data record may have session time duration data (e.g. an Acct-Session-Time attribute) missing, contain a zero value, contain an inaccurate value for the session (e.g., reporting a session as being 4 minutes long when it was in fact 3 minutes long), contain an unreasonably large value or is invalid as defined by RFC 2139, section 5.7 and so on. Invalid access time duration may occur, for example, when a modem bank of a service provider does not report disconnection by the user and the NAS 532 continues to accumulate session time until another session starts on the same physical modem port or a timeout occurs for some other reason.

[000178] Accounting/transaction data records with a duplicate invalid session-length can arise for a variety of reasons, for example:

Missing Acct-Session-Time

[000179] When an accounting/transaction data record is received by the netserver 470 and is missing the Acct-Session-Time attribute, the netserver 470 typically sends on an accounting/transaction data record with a zero session length.

Inaccurate Acct-Session-Time

[000180] Inaccurate time accounting by the NAS 532 or intentional fraud by a service provider can generate accounting/transaction data records with inaccurate session durations.

Acct-Session-Time of Zero

[000181] When an accounting/transaction data record with a zero value Session-Time attribute is received by the netserver 470, the netserver 470 typically sends on an accounting/transaction data record with a zero session length.

Large Acct-Session-Time

[000182] Due to fraud, malfunction or inappropriate configuration, session time accounting may identify sessions of extravagant duration.

Disconnect Detection Failure

[000183] "Long" sessions or multiple sessions with identical duration are sometimes due to malfunction and/or inappropriate configuration of the modem bank of the service provider, which fails to detect user, disconnect for extended periods.

Fraudulent Access

[000184] Extended session times may also be due to continuous use by malicious users.

Corrupted Acct-Session-Time

[000185] Errors in the field handling by the NAS 532, the authorization/authentication server 600 of the service provider, or the netserver 470 of the service provider may corrupt the session time attribute of an accounting/transaction data record. Based on actual samples, this occurs often when long, vendor-specific data are present in some preceding RADIUS packet.

Genuine Long Sessions

[000186] The accuracy of the filtering of long sessions is dependent upon the filter threshold (typically about 100 hours).

[000187] Enhanced accuracy may be achieved by correlating, associating or the like the session length provided in the SQM records with session length provided in the accounting records. As each transaction data record has its unique session identification, the session length may be obtained from an associated record using the session_id field of the acct_trans tables and session_id field of the SQM tables 860. Data missing in one transaction data record can thus be obtained from another transaction data record bearing the unique session identification.

Overlapping accounting records

[000188] In certain circumstances, transaction data records are received from service providers that include the same user credentials (e.g. the same user name and password) that overlap in time. In the present embodiment of the invention, as each access session includes a unique session identification, analysis of overlapping transaction data records may be facilitated. In particular, the session_id field of the acct_trans table 854 and the session_id field of the SQM table 860 may be used for determining the session details of these records. For example, using the unique session identification, it may be determined if such sessions are two genuine different sessions or if the sessions are generated due to a faulty NAS 532.

Disputed Records

[000189] As the session identification uniquely identifies each single access session, and data related to the session including the unique session identification is sent to various different servers in the system, dispute resolution may be facilitated. In particular, a customer support team could compare the session details in the authentication or authorized transactions, accounting and SQM tables 856, 854 and 860 respectfully thereby providing three different sources of transaction data uniquely associated with a single user access session. The session_id field of the tables facilitates association, correlation or the like to corroborate details of the particular user access session.

Challenge Provider Records Recording Quality

[000190] As each session is uniquely identified, and various different servers communicate transaction data records independently to the settlement system 474, the quality of transaction data records received from a particular service provider may be evaluated by comparing transaction data records from that particular service provider with records from other sources. This may assist a network access team in isolating problems that relate to the accounting function as such or relate to technology problems at the service provider.

Legitimate Id Usage By More Than One Person

[000191] As the unique session identification uniquely identifies each single user session, it may be used to identify separate user access occurrences in which legitimate use of a single user identification data is used by more that one user. Thus, 456 customers may share the same login name, e.g. because the organization is small and/or chooses to operate in such a fashion. In such instances, it is possible to see logins from multiple locations with coincidental start times and session lengths being the same. The inclusion of the unique session identification is used to investigate these situations with enhanced accuracy.

Policy Management Versioning Of Dialer

[000192] The unique session identification can be used to associate, correlate, or the like SQM records with accounting records whereby accounting records created without SQM records can be used to reveal the use of connection application (e.g., the connection application 466) technology, or non-supported versions thereof, that are not associated or provided by the access broker system 454. Typically, a report is created of customers and individual users who are using non-supported versions of the connection application 466 (e.g., connect dialer technology) thereby to help to migrate such users to a more current version. In certain circumstances, when an inappropriate connection application 466 is user by a customer 456, an account associated with the customer may be automatically disabled. Accordingly, the customer 456 may then be forced to contact the access brokerage system 454 to identify the problem and thereby force a version migration.

Overall Billing Process Quality Improvement

[000193] It will thus be appreciated that the inclusion of the unique session identification, that uniquely identifies all transaction data records associated with a single user session, enhances the accuracy of transaction record processing. Accordingly, less billing disputes are likely to arise and any dispute resolution that may arise may be settled more expeditiously.

[000194] Figure 13 is a diagram of a computer system 700, which may be configured as a network access device, such as 205, 305 or 405, a network dialer 466, or a netserver, such as 240, 350, 468, or 472. Computer system 700 includes a processor 750 operatively connected to random access memory (RAM) 735 and read only memory (ROM) 740 via a system bus 745. A processor 750 is also connected to a fixed disk 720, which may be an optical disk or other storage medium, through an input/output bus 730. Alternatively, the processor 750 may be connected to multiple storage devices through the input/output bus 730. The processor 750 communicates data using the system bus 745 and the input/output bus 730.

[000195] The system bus 745 and the input/output bus 730 may also receive inputs from a keypad 725 or an input device 710. The system bus 745 and the input/output bus 730 may provide outputs to a display 705, the fixed disk 720, and/or the output device 715. The memory and storage media 735, 740 may also include a flash memory, EEPROM, or any combination of the above.

[000196] The computer system 700 may be controlled by operating system software, which includes a file management system, such as, a disk operating system, which is part of the operating system software. The file management system may be stored in non-volatile storage device, such as the ROM 740, and may be configured to cause the processor 750 to execute the various functions required by the operating system to input and output data and to store data in the RAM 735 and on the ROM 740. For one embodiment of exemplary computer system 700, instructions may be stored on the fixed disk 720 or in the ROM 740 that cause the processor 750 to perform the functions of a network access device, such as the network access device 205 or 305. In an alternative embodiment, instructions may be stored on the fixed disk 720 or the ROM 740 that cause the processor 750 to perform the functions of a network decryption server, such as the netservers 240, 350, 472 or 468.

[000197] Thus, a method and system for securely authenticating network credentials or user data is described. In the foregoing detailed description, the invention has been described with reference to specific exemplary embodiments

thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader scope and spirit of the invention as set forth in the appended claims. The specification and drawings are, accordingly, to be regarded in an illustrative sense rather than a restrictive sense.

CLAIMS

What is claimed is:

1. A method to securely authenticate user credentials, the method including:

encrypting a user credential with a public key at an access device, the public key being part of a public/private key pair suitable for use with an encryption algorithm;

transmitting the encrypted network user credential from the access device to a decryption server;

decrypting the user credential at the decryption server with a private key, the private key being part of the public/private key pair suitable for use with the encryption algorithm; and

transmitting the decrypted user credential from the decryption server to an authentication server for verification.

2. The method of claim 1, in which the decryption server forms part of a multi-party service access environment including a plurality of access providers, the method including transmitting the user credential over a secure communication channel and decrypting the user credential of a user proximate an access provider associated with the user credential.

3. The method of claim 2, in which the access provider associated with the user credential is a customer of the multi-party access system, the user requesting access to a computer system of the customer via at least one service provider.

4. The method of claim 1, which includes generating the public/private key pair with an encryption algorithm suitable for use with elliptic curve cryptography.

5. The method of claim 1, wherein the encrypting includes encrypting a password input by a user with the public key.

6. The method of claim 1, wherein the encrypting includes encrypting a non-reversible hash of a password with the public key.

7. The method of claim 1, which includes:

transmitting the encrypted user credential from the access device to a network access server; and

transmitting the encrypted user credential from the network access server to the decryption server.

8. The method of claim 7, which includes:

negotiating with the network access server for use of an authentication protocol for transmitting the encrypted user credential from the access device to the network access server;

transmitting the encrypted user credential from the access device to the network access server using the negotiated authentication protocol; and

transmitting the encrypted user credential from the network access server to the decryption server.

9. The method of claim 8, which includes retrieving the private key from a private key database at the decryption server based on a username received from the access device.

10. The method of claim 1, wherein the user credential is a password authenticated using at least one of Point-to-Point protocol (PPP), Password Authentication Protocol (PAP), Challenge-Handshake Authentication Protocol (CHAP), Remote Authentication Dial In User Service (RADIUS) protocol, Terminal Access Controller Access Control System (TACACS) protocol, Lightweight Directory Access Protocol (LDAP), NT Domain authentication

protocol, Unix password authentication protocol, HyperText Transfer Protocol (HTTP), HyperText Transfer Protocol over Secure sockets layer (HTTPS), Extended Authentication Protocol (EAP), Transport Layer Security (TLS) protocol, Token Ring protocol and Secure Remote Password protocol (SRP).

11. The method of claim 10, wherein the encrypted user credential is at least substantially defined by standard characters described in RFC 2486.

12. The method of claim 11, wherein the encrypted user credentials that are transmitted to the decryption server are generated by a symbol transformation scheme of values generated by the encryption algorithm and characters described in RFC 2486.

13. The method of claim 1, which includes incrementing a counter each time authentication is requested and including a count of the counter with the user credential.

14. The method of claim 13, which includes including an access device identification with the user credential.

15. The method of claim 14, wherein the access device identification uniquely identifies a connection application via which authentication is requested.

16. The method of claim 15, which includes generating a checksum character by generating the MD5 hash of the count, access device identification and a point of the encryption algorithm.

17. The method of claim 16, wherein standard characters are added to a byte of a random point generated by an elliptic curve cryptography algorithm where after a modulus 95 function is performed.

18. A method of authenticating user data of a user requesting access to a service access system including a plurality of service providers, the method including:

encrypting the user data with a public key, the public key being part of a public/private key pair suitable for use with an encryption algorithm; and

transmitting the encrypted user data to a decryption server for decryption using the private key.

19. The method of claim 18, in which the encrypted user data is configured so that it can be transmitted using at least one of Point-to-Point protocol (PPP), Password Authentication Protocol (PAP), Challenge-Handshake Authentication Protocol (CHAP), Remote Authentication Dial In User Service (RADIUS) protocol, Terminal Access Controller Access Control System (TACACS) protocol, Lightweight Directory Access Protocol (LDAP), NT Domain authentication protocol, Unix password authentication protocol, HyperText Transfer Protocol (HTTP), HyperText Transfer Protocol over Secure sockets layer (HTTPS), Extended Authentication Protocol (EAP), Transport Layer Security (TLS) protocol, Token Ring protocol and Secure Remote Password protocol (SRP).

20. The method of claim 19, in which the encrypted user data is modified prior to transmission to conform to RFC 2486 standards using character transformation.

21. The method of claim 20, which includes encrypting a non-reversible hash of a user password with the public key.

22. The method of claim 21, in which the encrypted user data is modified prior to transmission to include only plain text ASCII characters.

23. The method of claim 22, in which the encrypted data is included in at least one of a user password field and a user identification field of the protocol.

24. The method of claim 23, in which user data for inclusion in the password field is encrypted with a random point generated using an elliptic curve cryptography algorithm.

25. The method of claim 18, in which the user requests access to the network via a connect dialer, the method including:

incrementing a counter of the connect dialer, the count of the counter identifying a user session for which the user requires authentication;

retrieving a connect dialer identification that identifies the dialer; and

including the count and dialer identification with the encrypted data prior to transmission thereof to the service access system.

26. The method of claim 25, which includes:

generating a checksum from the count and connect dialer identification; and

encrypting the checksum using a random point of an elliptic curve cryptography algorithm.

27. The method of claim 26, which includes encoding 7 bits with a single byte from the random point.

28. The method of claim 26, which includes:

concatenating an encoded user password, an encrypted and encoded x coordinate of the random point with encoded checksum bits to define encrypted credentials; and

transmitting the encrypted credentials in user password and identification fields provided by a standard authentication protocol.

29. The method of claim 25, in which includes encrypting a non-reversible hash of a user password with the public key.

30. The method of claim 25, which includes negotiating with a network access server for use of an authentication protocol for transmitting the network user credential from the network access device to the network decryption server.

31. A method of authenticating user data of a user requesting access to a service access system including a plurality of service providers, the method including:

receiving encrypted user data from an access device;

decrypting the encrypted user data using a private key; and

transmitting the decrypted user data to an authentication server for authentication.

32. The method of claim 31, in which the encrypted user data is received from the access device via at least one service provider.

33. The method of claim 31, which includes extracting encrypted user data from at least one of a user password and a user identification field of a an authentication protocol selected from at least one of Point-to-Point protocol (PPP), Password Authentication Protocol (PAP), Challenge-Handshake Authentication Protocol (CHAP), Remote Authentication Dial In User Service (RADIUS) protocol, Terminal Access Controller Access Control System (TACACS) protocol, Lightweight Directory Access Protocol (LDAP), NT Domain authentication protocol, Unix password authentication protocol, HyperText Transfer Protocol (HTTP), HyperText Transfer Protocol over Secure

sockets layer (HTTPS), Extended Authentication Protocol (EAP), Transport Layer Security (TLS) protocol, Token Ring protocol and Secure Remote Password protocol (SRP).

34. The method of claim 33, in which includes:

receiving an encrypted random point from the access device, the random point being generated in response to encryption based on elliptic curve cryptography; and

decrypting the encrypted user data using the random point and the private key.

35. The method of claim 34, which includes applying symbol transformation to obtain the random point.

36. The method of claim 34, which includes retrieving the private key from a private key database provided at the decryption server based on a username received from the access device.

37. The method of claim 31, which includes:

identifying a count and an access device identification in the encrypted user data;

comparing the decrypted count and access device identification with a reference count and an access device identification; and

selectively rejecting the access request based on the comparison.

38. A computer readable medium, having stored thereon:

a first sequence of instructions which, when executed by a processor, causes the processor to encrypt user data with a public key, the public key being part of a public/private key pair suitable for use with an encryption algorithm; and

a second sequence of instructions which, when executed by a processor, causes the processor to transmit the encrypted user credential to a decryption server.

39. The computer readable medium of claim 38, wherein the first sequence of instructions, when executed by a processor, causes the processor to encrypt user data with the public key, the public key part of the public/private key pair suitable for use with the encryption algorithm based on elliptic curve cryptography.

40. The computer readable medium of claim 38, wherein the first sequence of instructions, when executed by a processor, causes the processor to encrypt a password input by a user with the public key.

41. The computer readable medium of claim 38, wherein the first sequence of instructions, when executed by a processor, causes the processor to encrypt a non-reversible hash with the public key.

42. The computer readable medium of claim 38, wherein the first sequence of instructions, when executed by a processor, causes the processor to negotiate with a network access server the use of an authentication protocol for transmitting the encrypted user data to the network access server.

43. A computer readable medium, having stored thereon:

a first sequence of instructions which, when executed by a processor, causes the processor to receive encrypted user data from an access device;

a second sequence of instructions which, when executed by a processor, causes the processor to decrypt the encrypted user data using a private key, the private key being suitable for use with an encryption algorithm; and

a third sequence of instructions which, when executed by a processor, causes the processor to transmit the decrypted user data to an authentication server for verification.

44. The computer readable medium of claim 43, wherein the second sequence of instructions, when executed by a processor, causes the processor to decrypt the encrypted user data using the private key, the private key generated utilizing the encryption algorithm based on elliptic curve cryptography.

45. The computer readable medium of claim 43, including a fourth sequence of instructions which, when executed by a processor, causes the processor to retrieve the private key from a private key database based on a username received from the access device.

46. A computer to authenticate user data of a user requesting access to a service access system including a plurality of service providers, the computer including:

a receiver to receive encrypted user data from an access device;

decryptor to decrypt the encrypted user data using a private key; and

a transmitter to transmit the decrypted user data to an authentication server for authentication.

47. The computer of claim 46, which includes a processor to extract encrypted user data from at least one of a user password and a user identification field of a authentication protocol selected from at least one of Point-to-Point protocol (PPP), Password Authentication Protocol (PAP), Challenge-Handshake Authentication Protocol (CHAP), Remote Authentication Dial In User Service (RADIUS) protocol, Terminal Access Controller Access Control System (TACACS) protocol, Lightweight Directory Access Protocol

(LDAP), NT Domain authentication protocol, Unix password authentication protocol, HyperText Transfer Protocol (HTTP), HyperText Transfer Protocol over Secure sockets layer (HTTPS), Extended Authentication Protocol (EAP), Transport Layer Security (TLS) protocol, Token Ring protocol and Secure Remote Password protocol (SRP).

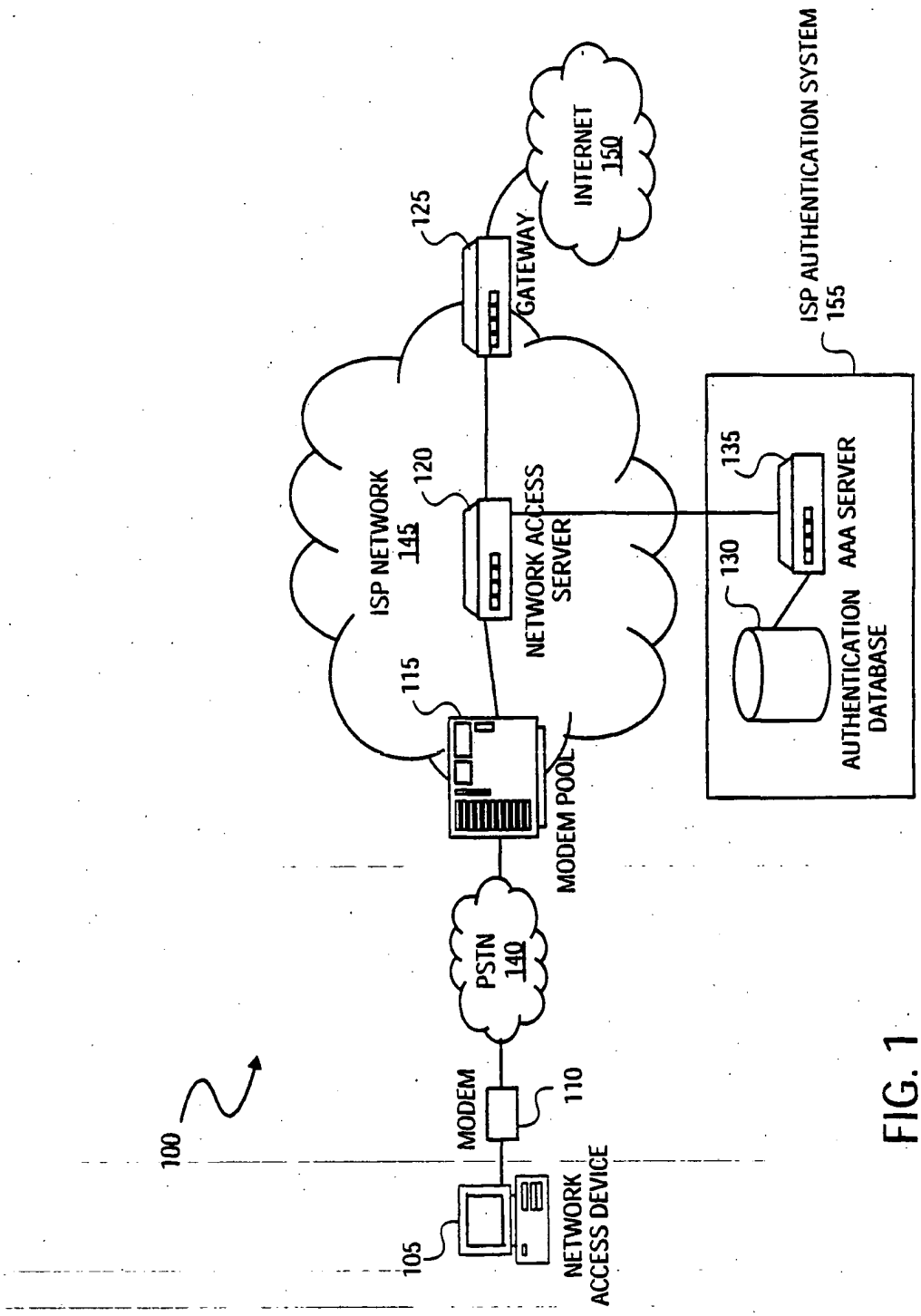


FIG. 1
(PRIOR ART)

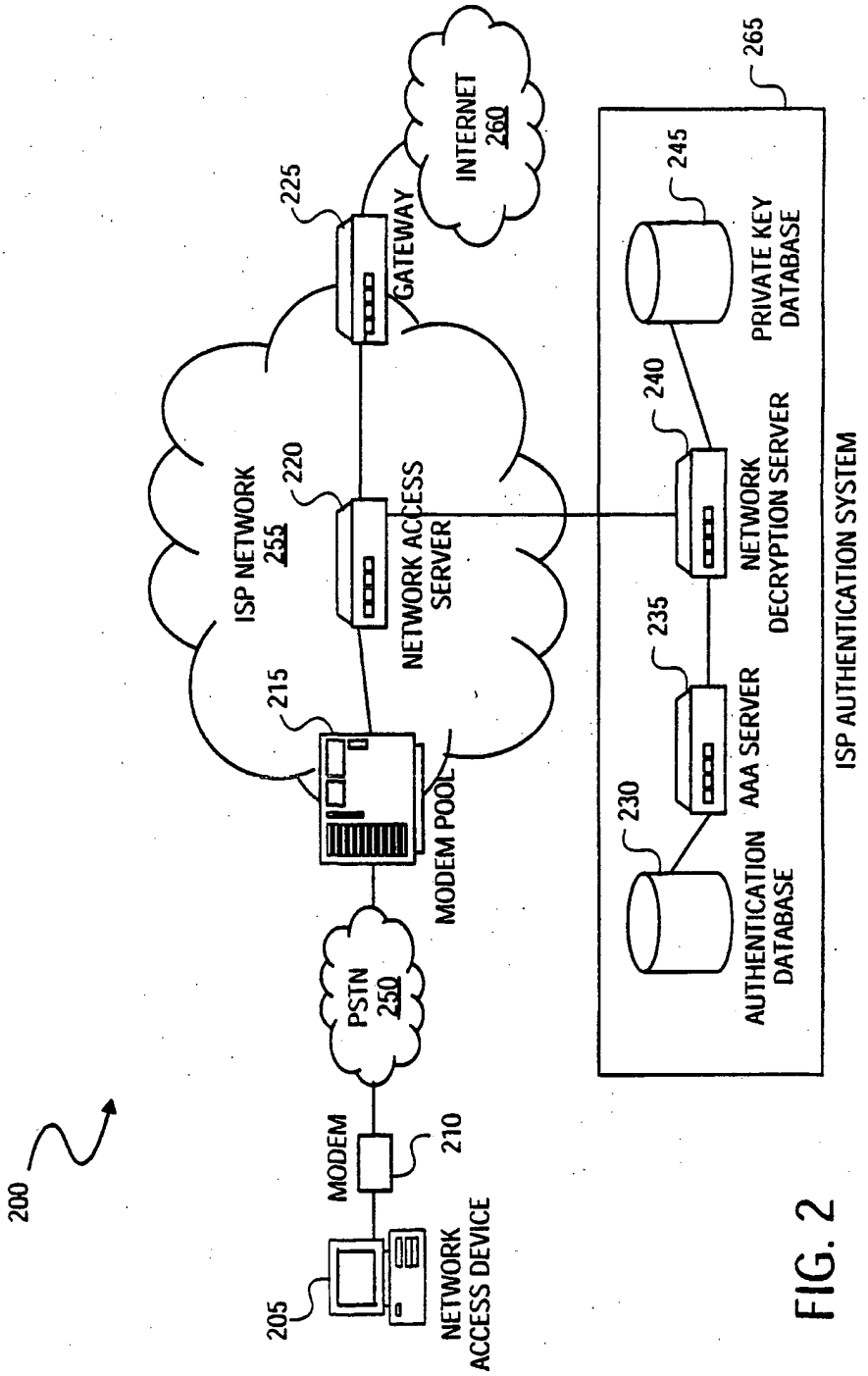

200
205
NETWORK ACCESS DEVICE
210
MODEM
PSTN 250
215
MODEM POOL
ISP NETWORK 255
220
NETWORK ACCESS SERVER
225
GATEWAY
INTERNET 260
265
ISP AUTHENTICATION SYSTEM
230
AUTHENTICATION DATABASE
235
AAA SERVER
240
NETWORK DECRYPTION SERVER
245
PRIVATE KEY DATABASE

FIG. 2

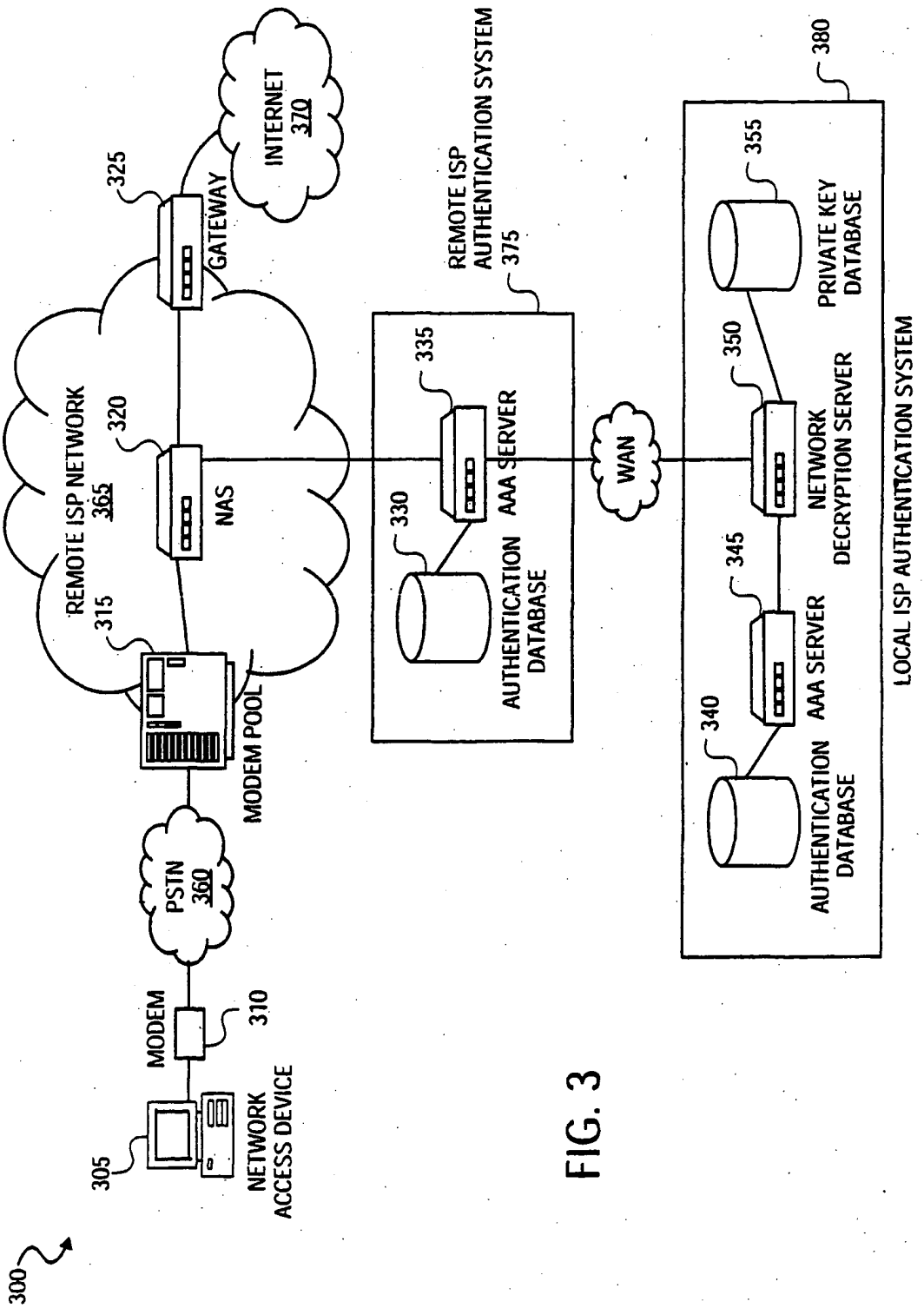

300
305
NETWORK ACCESS DEVICE
MODEM
310
PSTN 360
MODEM POOL
315
REMOTE ISP NETWORK 365
320
NAS
325
GATEWAY
INTERNET 370
330
AUTHENTICATION DATABASE
335
AAA SERVER
REMOTE ISP AUTHENTICATION SYSTEM 375
WAN
340
AUTHENTICATION DATABASE
345
AAA SERVER
350
NETWORK DECRYPTION SERVER
355
PRIVATE KEY DATABASE
380
LOCAL ISP AUTHENTICATION SYSTEM

FIG. 3

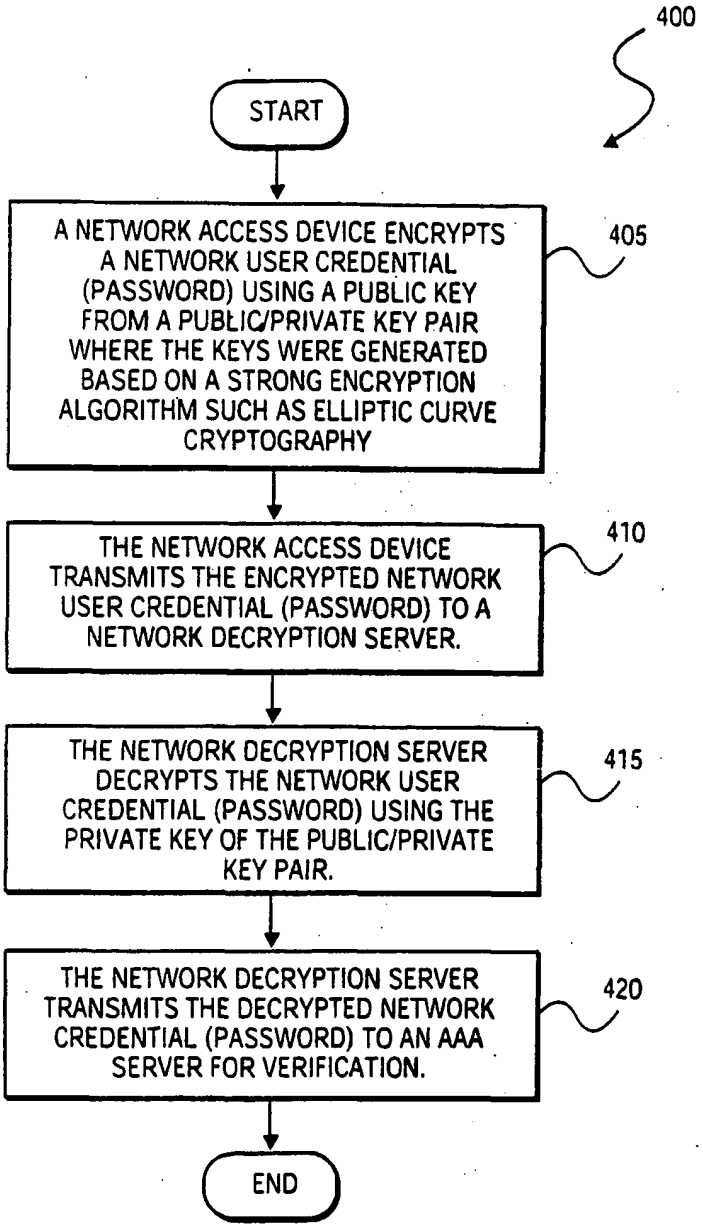

400
START
A NETWORK ACCESS DEVICE ENCRYPTS A NETWORK USER CREDENTIAL (PASSWORD) USING A PUBLIC KEY FROM A PUBLIC/PRIVATE KEY PAIR WHERE THE KEYS WERE GENERATED BASED ON A STRONG ENCRYPTION ALGORITHM SUCH AS ELLIPTIC CURVE CRYPTOGRAPHY
405
THE NETWORK ACCESS DEVICE TRANSMITS THE ENCRYPTED NETWORK USER CREDENTIAL (PASSWORD) TO A NETWORK DECRYPTION SERVER.
410
THE NETWORK DECRYPTION SERVER DECRYPTS THE NETWORK USER CREDENTIAL (PASSWORD) USING THE PRIVATE KEY OF THE PUBLIC/PRIVATE KEY PAIR.
415
THE NETWORK DECRYPTION SERVER TRANSMITS THE DECRYPTED NETWORK CREDENTIAL (PASSWORD) TO AN AAA SERVER FOR VERIFICATION.
420
END

FIG. 4

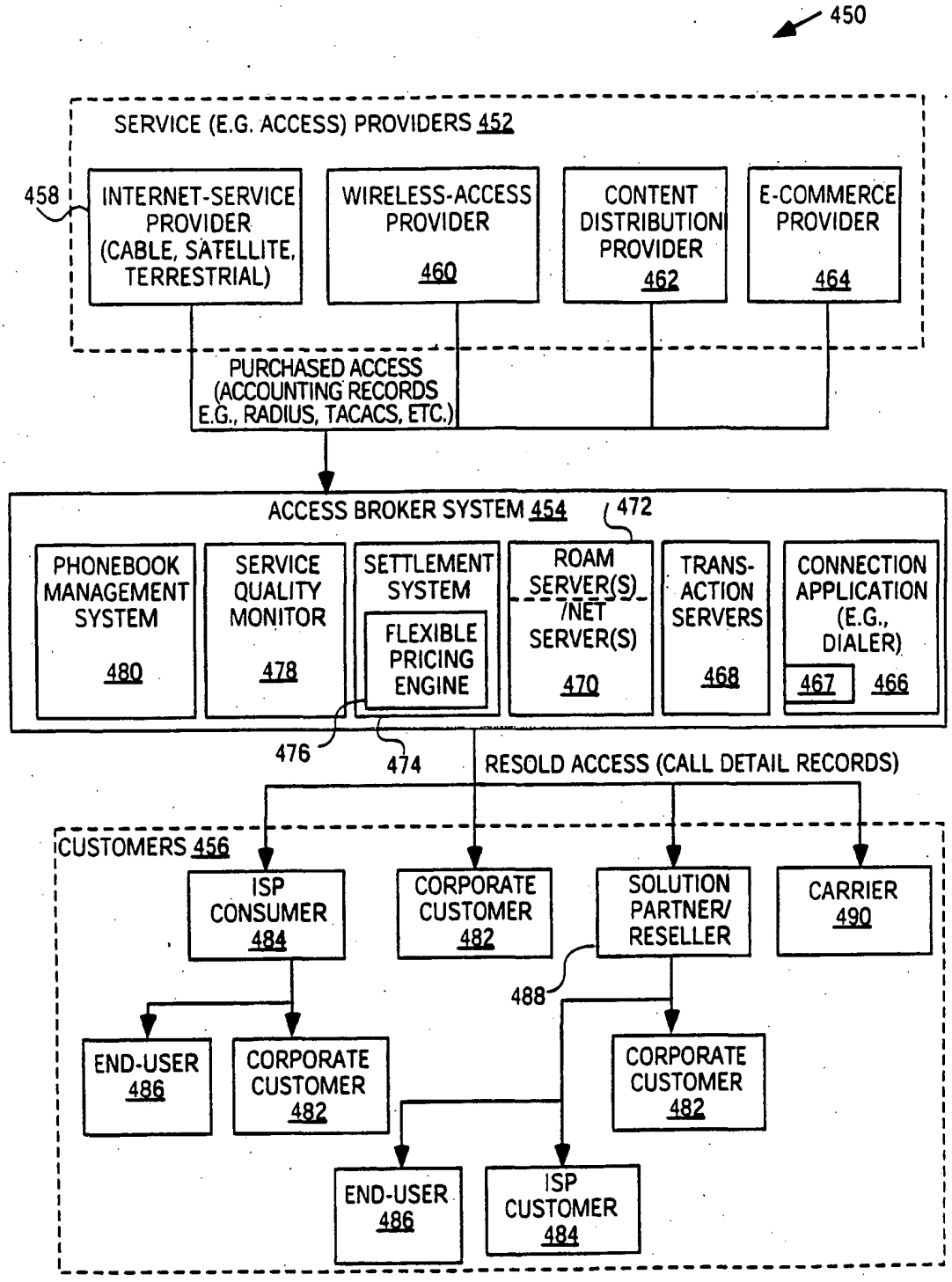

450
SERVICE (E.G. ACCESS) PROVIDERS 452
458
INTERNET-SERVICE PROVIDER (CABLE, SATELLITE, TERRESTRIAL)
WIRELESS-ACCESS PROVIDER 460
CONTENT DISTRIBUTION PROVIDER 462
E-COMMERCE PROVIDER 464
PURCHASED ACCESS (ACCOUNTING RECORDS E.G., RADIUS, TACACS, ETC.)
ACCESS BROKER SYSTEM 454
472
PHONEBOOK MANAGEMENT SYSTEM 480
SERVICE QUALITY MONITOR 478
SETTLEMENT SYSTEM
FLEXIBLE PRICING ENGINE
ROAM SERVER(S) /NET SERVER(S) 470
TRANS-ACTION SERVERS 468
CONNECTION APPLICATION (E.G., DIALER) 467 466
476
474
RESOLD ACCESS (CALL DETAIL RECORDS)
CUSTOMERS 456
ISP CONSUMER 484
CORPORATE CUSTOMER 482
SOLUTION PARTNER/ RESELLER
CARRIER 490
488
END-USER 486
CORPORATE CUSTOMER 482
CORPORATE CUSTOMER 482
END-USER 486
ISP CUSTOMER 484

FIG. 5

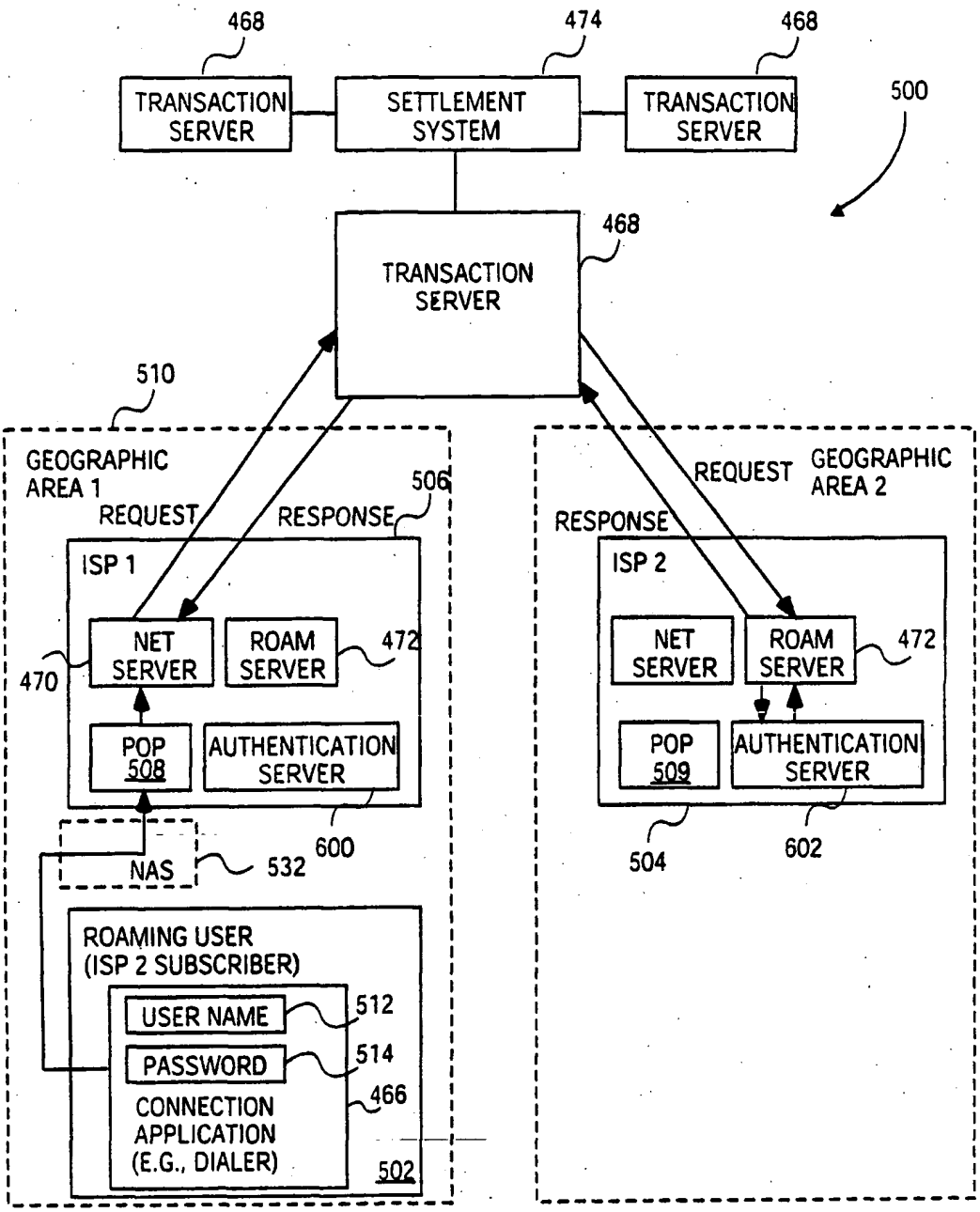

468
474
468
TRANSACTION SERVER
SETTLEMENT SYSTEM
TRANSACTION SERVER
500
468
TRANSACTION SERVER
510
GEOGRAPHIC AREA 1
REQUEST
RESPONSE
506
ISP 1
NET SERVER
ROAM SERVER
472
470
POP 508
AUTHENTICATION SERVER
600
NAS
532
ROAMING USER (ISP 2 SUBSCRIBER)
USER NAME
512
PASSWORD
514
CONNECTION APPLICATION (E.G., DIALER)
466
502
REQUEST
GEOGRAPHIC AREA 2
RESPONSE
ISP 2
NET SERVER
ROAM SERVER
472
POP 509
AUTHENTICATION SERVER
504
602

FIG. 6

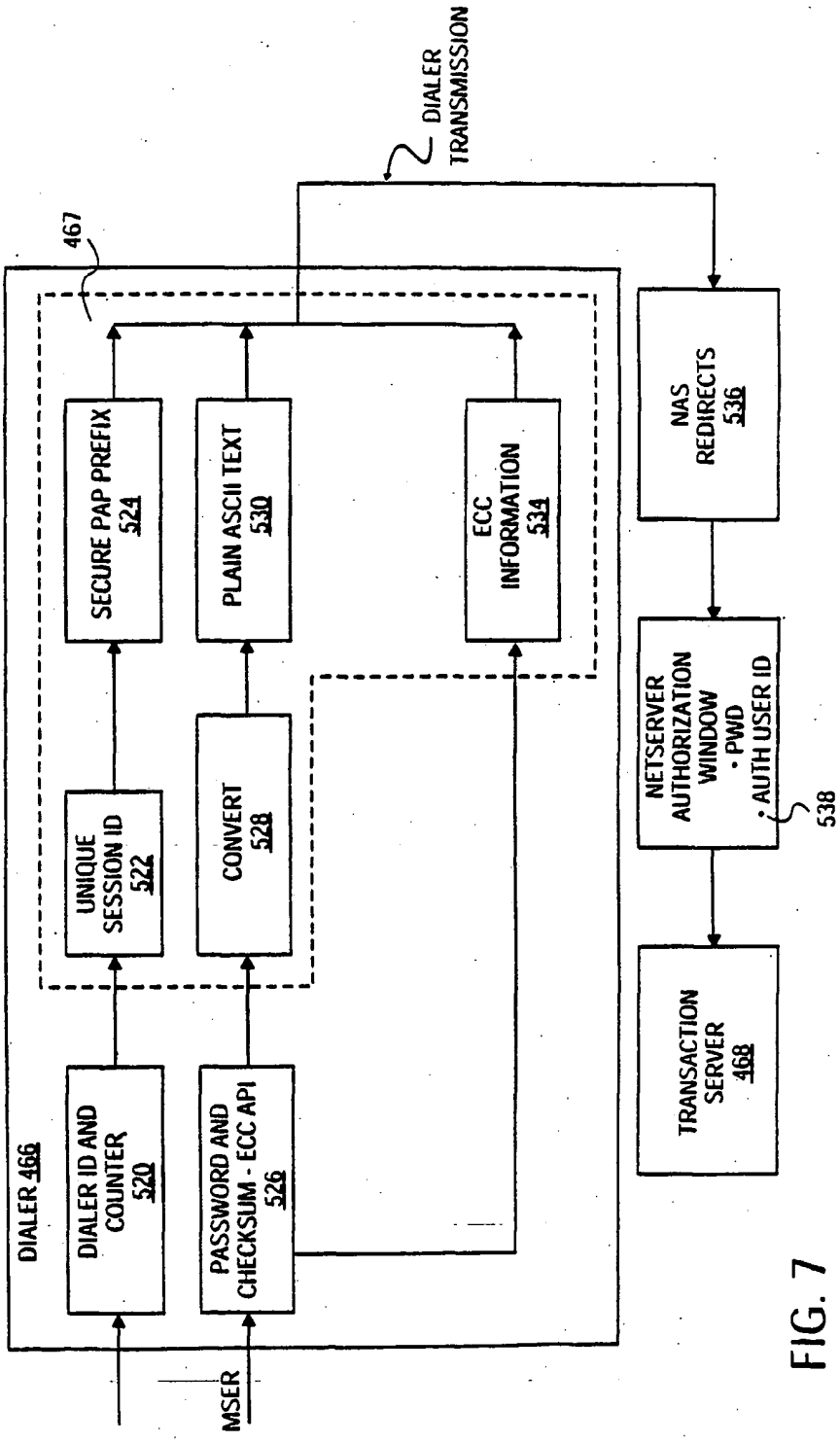

DIALER TRANSMISSION
467
SECURE PAP PREFIX 524
PLAIN ASCII TEXT 530
ECC INFORMATION 534
NAS REDIRECTS 536
UNIQUE SESSION ID 522
CONVERT 528
NETSERVER AUTHORIZATION WINDOW
· PWD
· AUTH USER ID
538
DIALER 466
DIALER ID AND COUNTER 520
PASSWORD AND CHECKSUM - ECC API 526
TRANSACTION SERVER 468
MSER

FIG. 7

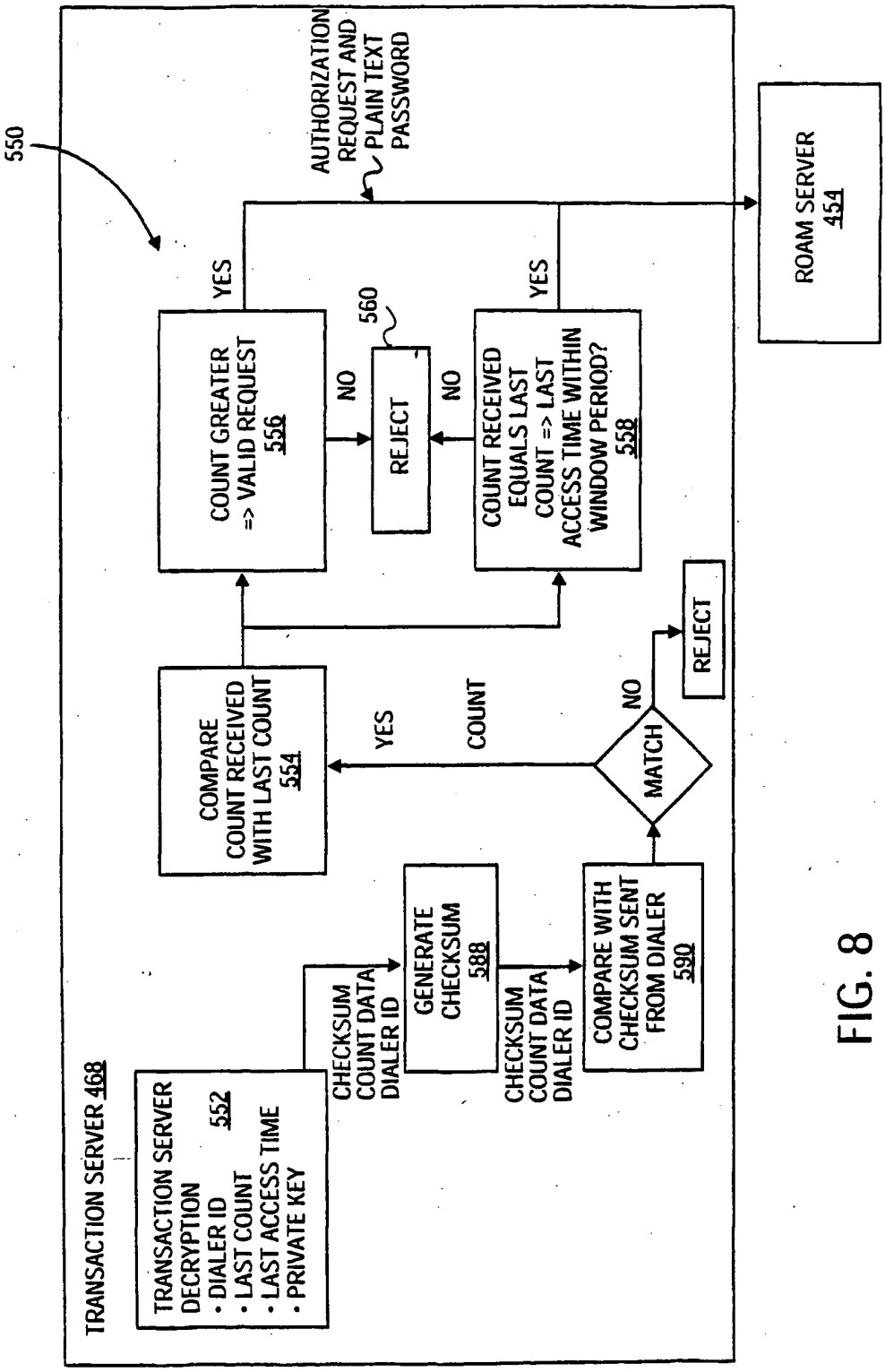

550
AUTHORIZATION REQUEST AND PLAIN TEXT PASSWORD
ROAM SERVER 454
TRANSACTION SERVER 468
TRANSACTION SERVER DECRYPTION
• DIALER ID 552
• LAST COUNT
• LAST ACCESS TIME
• PRIVATE KEY
CHECKSUM COUNT DATA DIALER ID
GENERATE CHECKSUM 588
CHECKSUM COUNT DATA DIALER ID
COMPARE WITH CHECKSUM SENT FROM DIALER 590
MATCH
NO
REJECT
YES
COUNT
COMPARE COUNT RECEIVED WITH LAST COUNT 554
COUNT GREATER => VALID REQUEST 556
NO
560
REJECT
YES
COUNT RECEIVED EQUALS LAST COUNT => LAST ACCESS TIME WITHIN WINDOW PERIOD? 558
NO
YES

FIG. 8

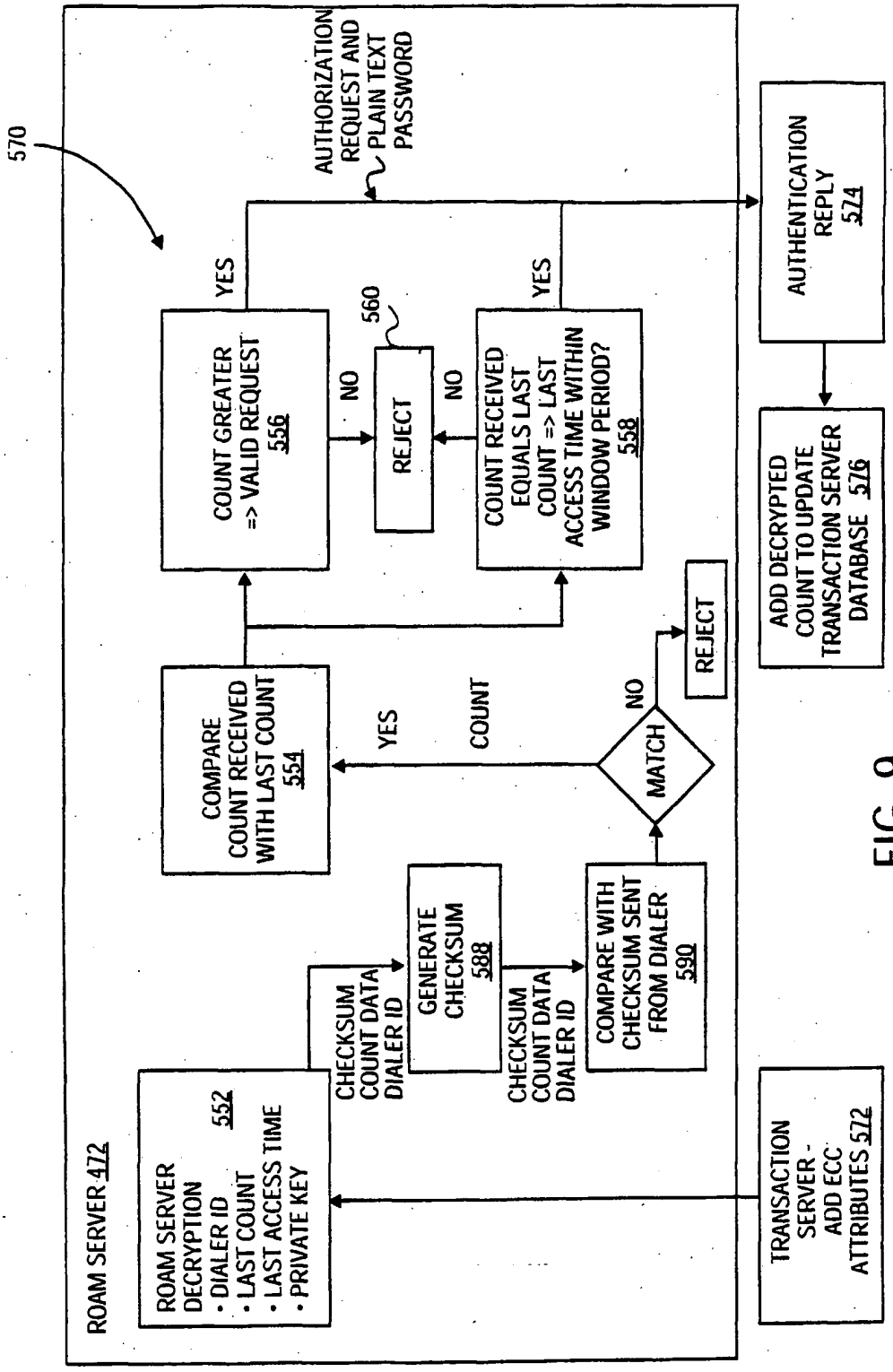

570
ROAM SERVER 472
ROAM SERVER DECRYPTION
• DIALER ID 552
• LAST COUNT
• LAST ACCESS TIME
• PRIVATE KEY
CHECKSUM COUNT DATA DIALER ID
GENERATE CHECKSUM 588
CHECKSUM COUNT DATA DIALER ID
COMPARE WITH CHECKSUM SENT FROM DIALER 590
MATCH
NO
REJECT
YES
COUNT
COMPARE COUNT RECEIVED WITH LAST COUNT 554
COUNT GREATER => VALID REQUEST 556
NO
REJECT 560
COUNT RECEIVED EQUALS LAST COUNT => LAST ACCESS TIME WITHIN WINDOW PERIOD? 558
YES
YES
AUTHORIZATION REQUEST AND PLAIN TEXT PASSWORD
AUTHENTICATION REPLY 574
ADD DECRYPTED COUNT TO UPDATE TRANSACTION SERVER DATABASE 576
TRANSACTION SERVER - ADD ECC ATTRIBUTES 572

FIG. 9

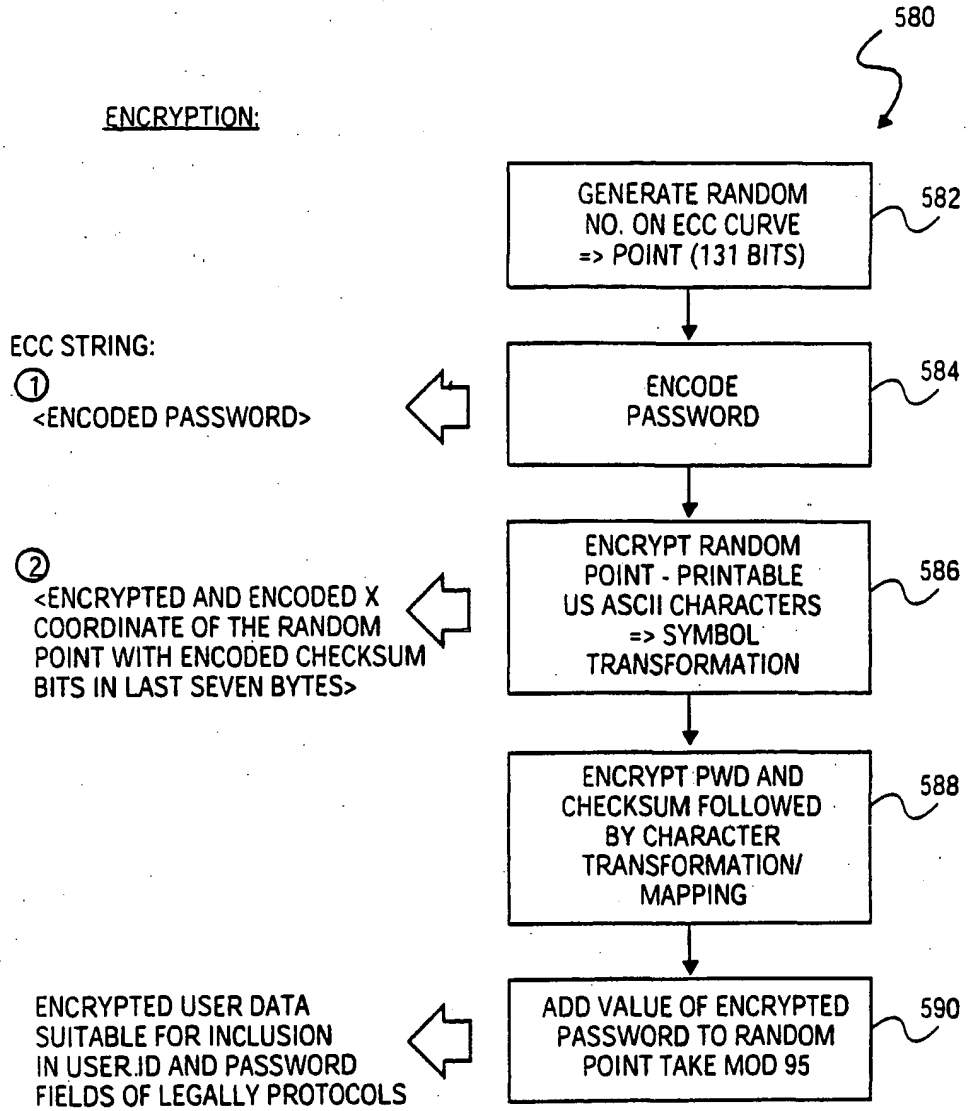

580
ENCRYPTION:
GENERATE RANDOM NO. ON ECC CURVE => POINT (131 BITS)
582
ECC STRING:
① <ENCODED PASSWORD>
ENCODE PASSWORD
584
② <ENCRYPTED AND ENCODED X COORDINATE OF THE RANDOM POINT WITH ENCODED CHECKSUM BITS IN LAST SEVEN BYTES>
ENCRYPT RANDOM POINT - PRINTABLE US ASCII CHARACTERS => SYMBOL TRANSFORMATION
586
ENCRYPT PWD AND CHECKSUM FOLLOWED BY CHARACTER TRANSFORMATION/ MAPPING
588
ENCRYPTED USER DATA SUITABLE FOR INCLUSION IN USER.ID AND PASSWORD FIELDS OF LEGALLY PROTOCOLS
ADD VALUE OF ENCRYPTED PASSWORD TO RANDOM POINT TAKE MOD 95
590

FIG. 10

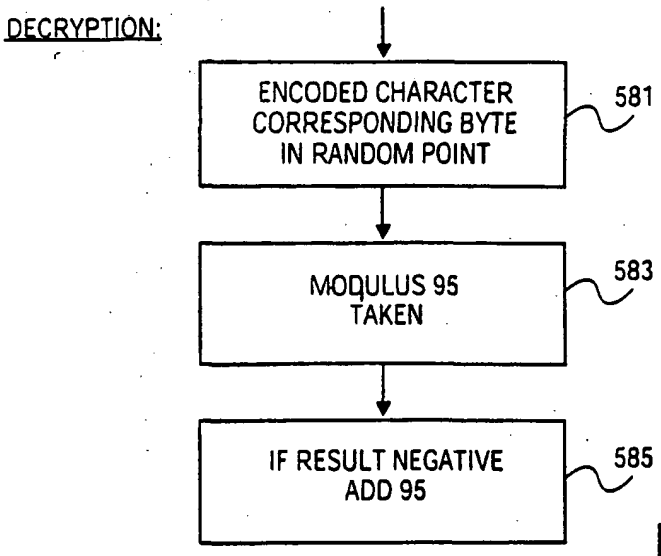

DECRYPTION:
ENCODED CHARACTER CORRESPONDING BYTE IN RANDOM POINT
581
MODULUS 95 TAKEN
583
IF RESULT NEGATIVE ADD 95
585

FIG. 11

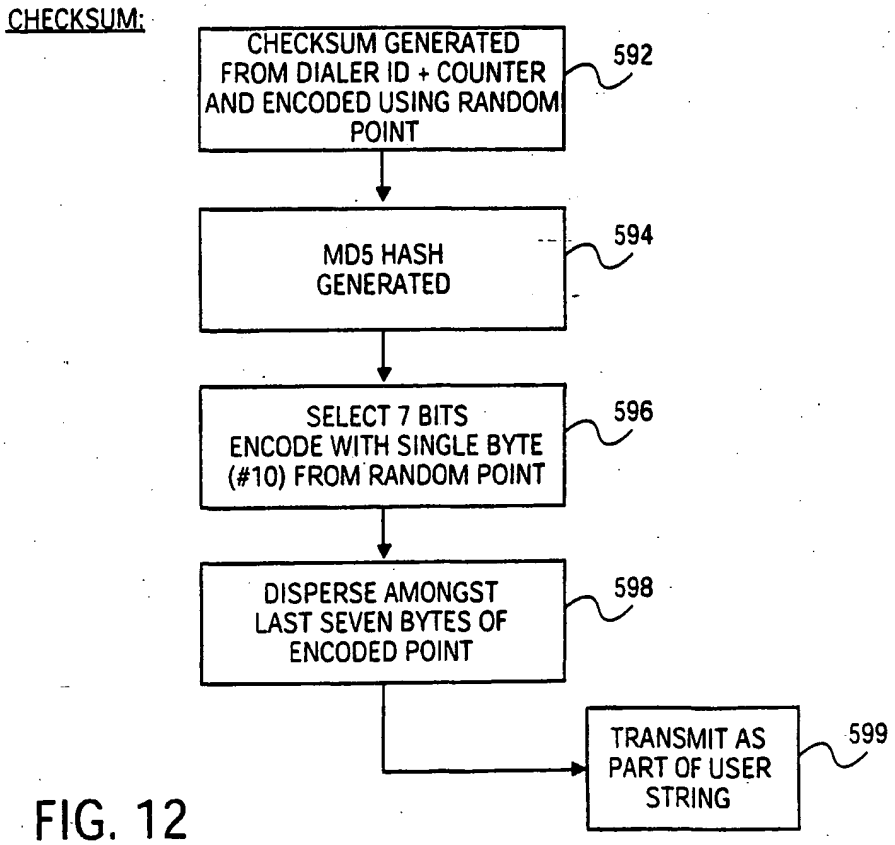

CHECKSUM:
CHECKSUM GENERATED FROM DIALER ID + COUNTER AND ENCODED USING RANDOM POINT
592
MD5 HASH GENERATED
594
SELECT 7 BITS ENCODE WITH SINGLE BYTE (#10) FROM RANDOM POINT
596
DISPERSE AMONGST LAST SEVEN BYTES OF ENCODED POINT
598
TRANSMIT AS PART OF USER STRING
599

FIG. 12

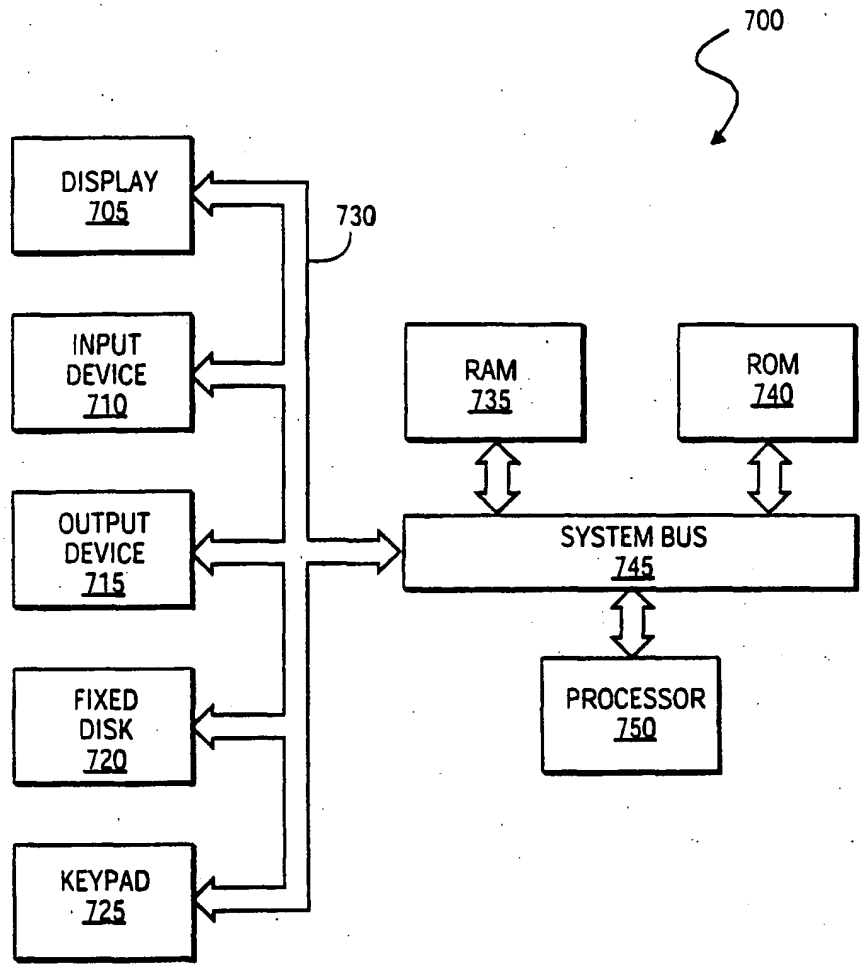

700
DISPLAY
705
730
INPUT
DEVICE
710
OUTPUT
DEVICE
715
FIXED
DISK
720
KEYPAD
725
RAM
735
ROM
740
SYSTEM BUS
745
PROCESSOR
750

FIG. 13

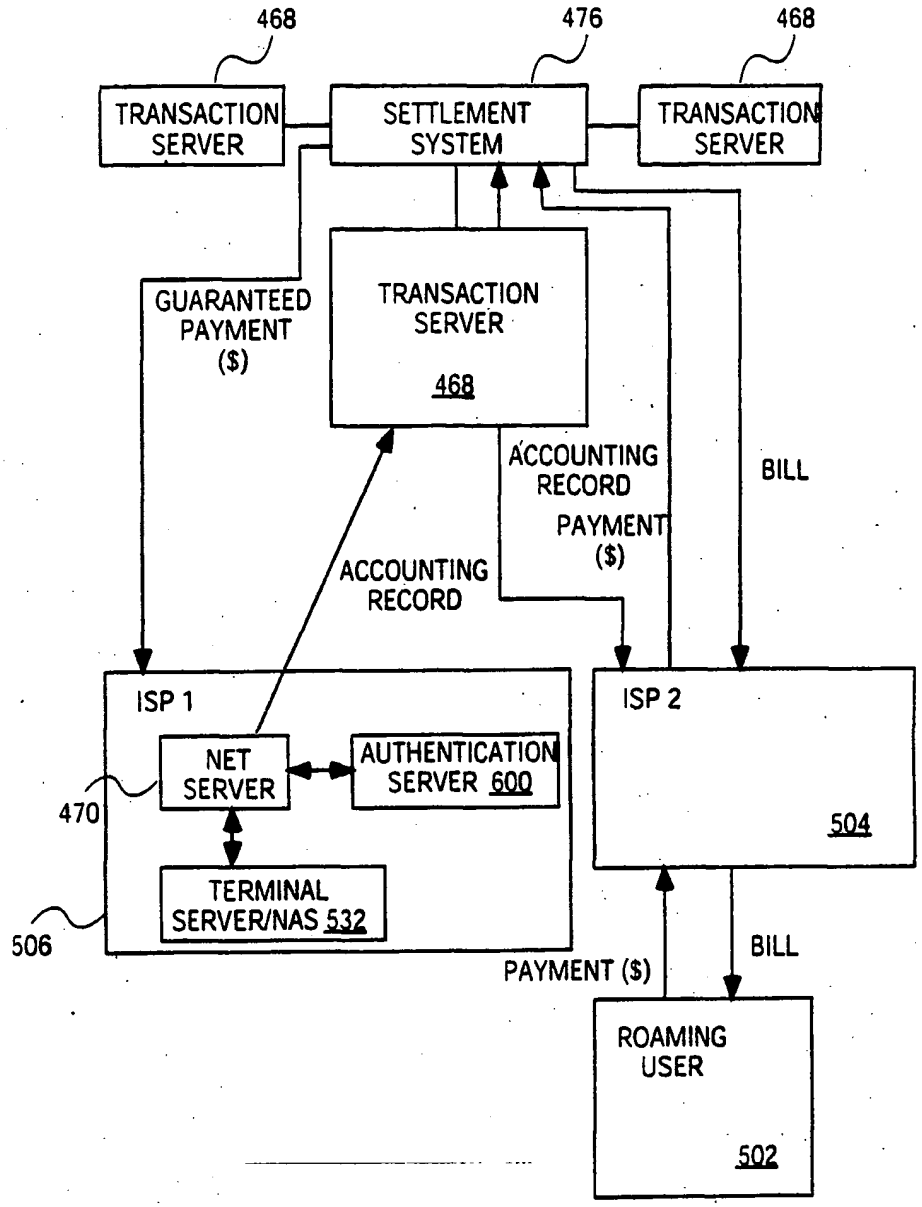

468
476
468
TRANSACTION SERVER
SETTLEMENT SYSTEM
TRANSACTION SERVER
GUARANTEED PAYMENT ($)
TRANSACTION SERVER 468
ACCOUNTING RECORD
PAYMENT ($)
BILL
ACCOUNTING RECORD
ISP 1
ISP 2
NET SERVER
AUTHENTICATION SERVER 600
470
504
TERMINAL SERVER/NAS 532
506
PAYMENT ($)
BILL
ROAMING USER
502

FIG. 14

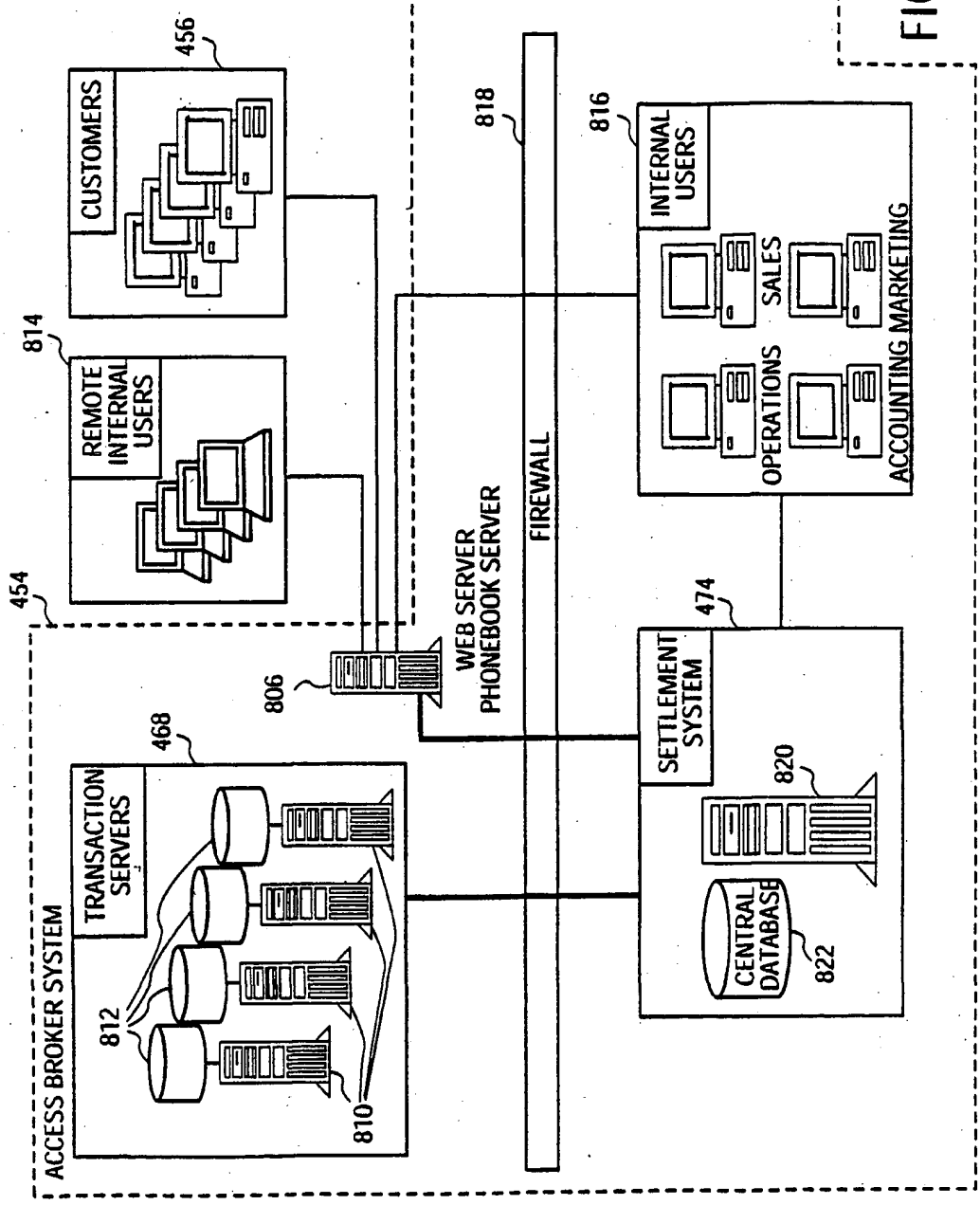

ACCESS BROKER SYSTEM
454
TRANSACTION SERVERS
468
812
810
806
WEB SERVER PHONEBOOK SERVER
FIREWALL
818
816
INTERNAL USERS
SALES
OPERATIONS
ACCOUNTING MARKETING
SETTLEMENT SYSTEM
474
820
CENTRAL DATABASE
822
CUSTOMERS
456
REMOTE INTERNAL USERS
814

FIG. 15

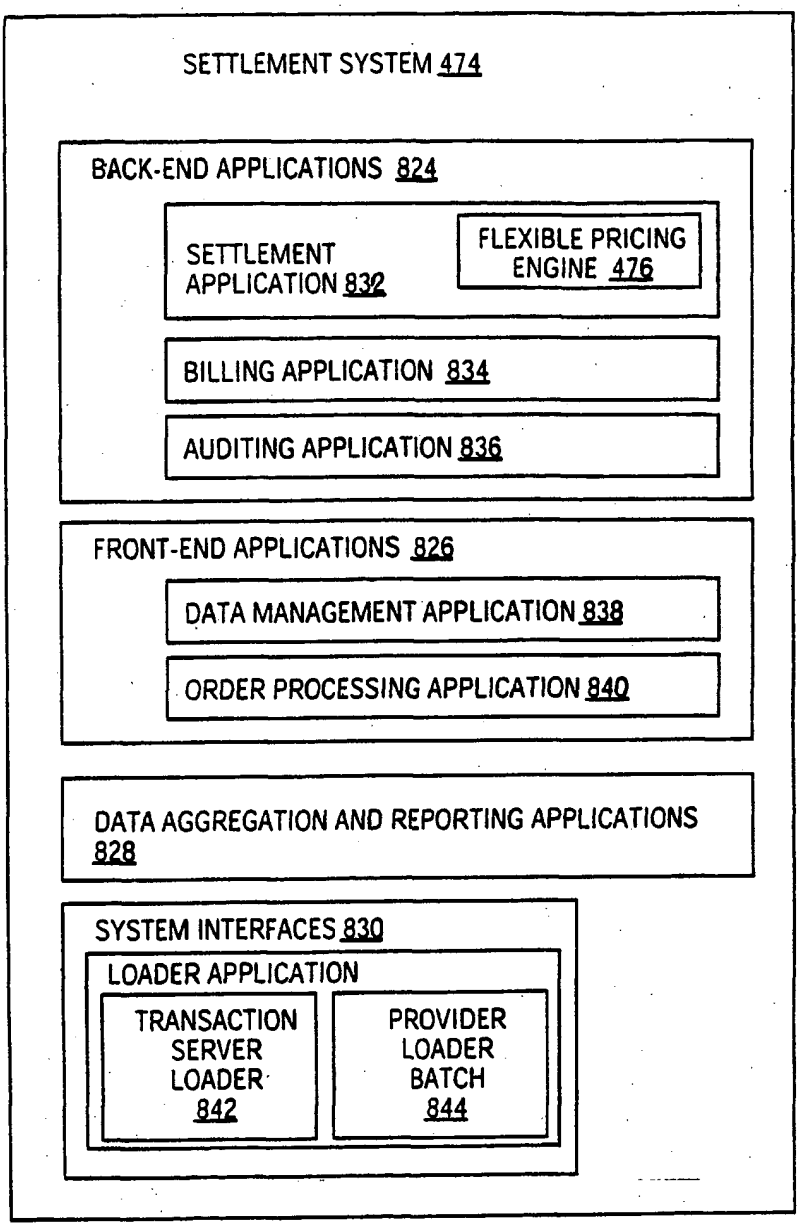

SETTLEMENT SYSTEM 474
BACK-END APPLICATIONS 824
SETTLEMENT APPLICATION 832
FLEXIBLE PRICING ENGINE 476
BILLING APPLICATION 834
AUDITING APPLICATION 836
FRONT-END APPLICATIONS 826
DATA MANAGEMENT APPLICATION 838
ORDER PROCESSING APPLICATION 840
DATA AGGREGATION AND REPORTING APPLICATIONS 828
SYSTEM INTERFACES 830
LOADER APPLICATION
TRANSACTION SERVER LOADER 842
PROVIDER LOADER BATCH 844

FIG. 16

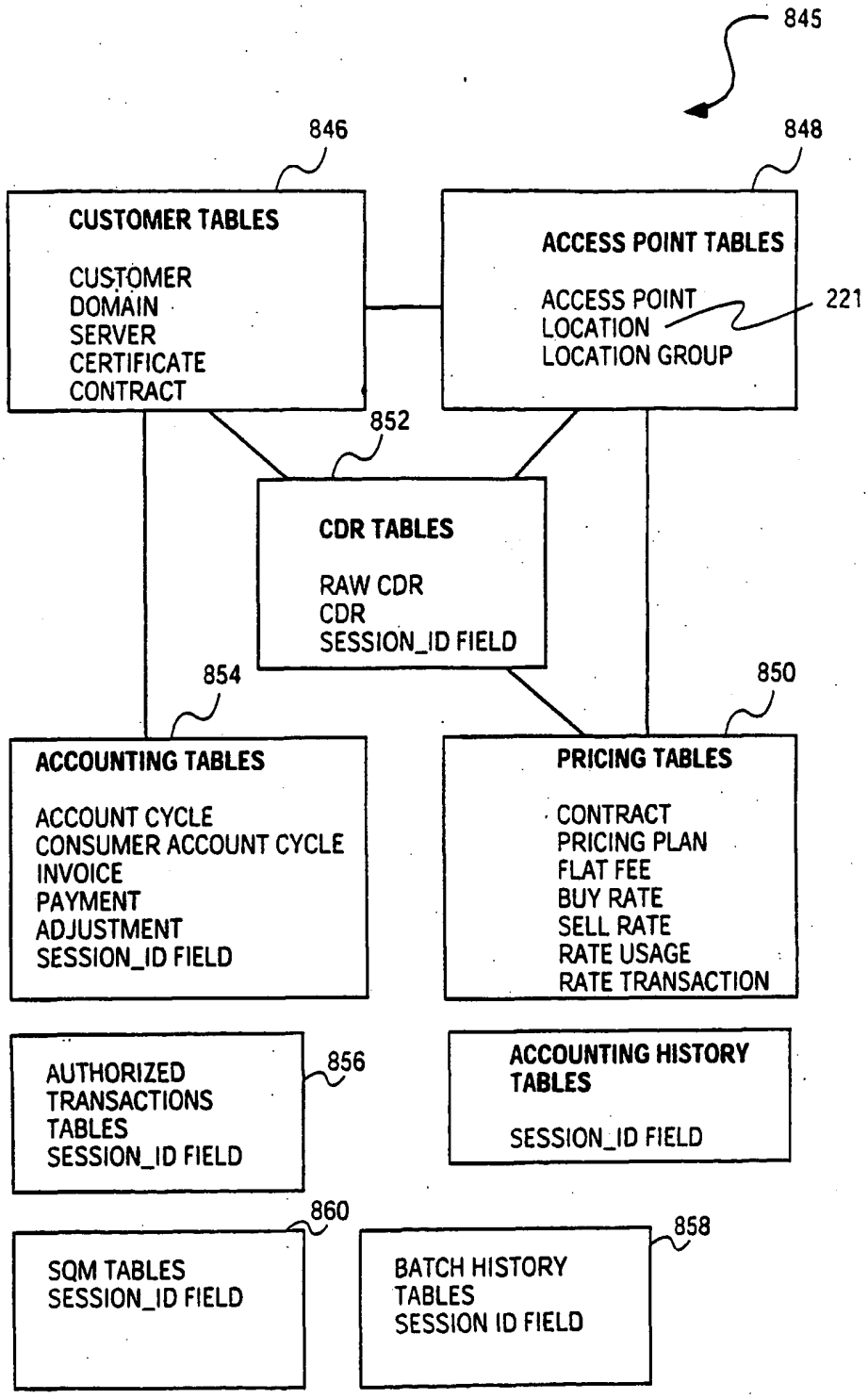

845
846
848
CUSTOMER TABLES
CUSTOMER
DOMAIN
SERVER
CERTIFICATE
CONTRACT
ACCESS POINT TABLES
ACCESS POINT
LOCATION
LOCATION GROUP
221
852
CDR TABLES
RAW CDR
CDR
SESSION_ID FIELD
854
850
ACCOUNTING TABLES
ACCOUNT CYCLE
CONSUMER ACCOUNT CYCLE
INVOICE
PAYMENT
ADJUSTMENT
SESSION_ID FIELD
PRICING TABLES
CONTRACT
PRICING PLAN
FLAT FEE
BUY RATE
SELL RATE
RATE USAGE
RATE TRANSACTION
AUTHORIZED
TRANSACTIONS
TABLES
SESSION_ID FIELD
856
ACCOUNTING HISTORY
TABLES
SESSION_ID FIELD
860
SQM TABLES
SESSION_ID FIELD
858
BATCH HISTORY
TABLES
SESSION ID FIELD

FIG. 17A

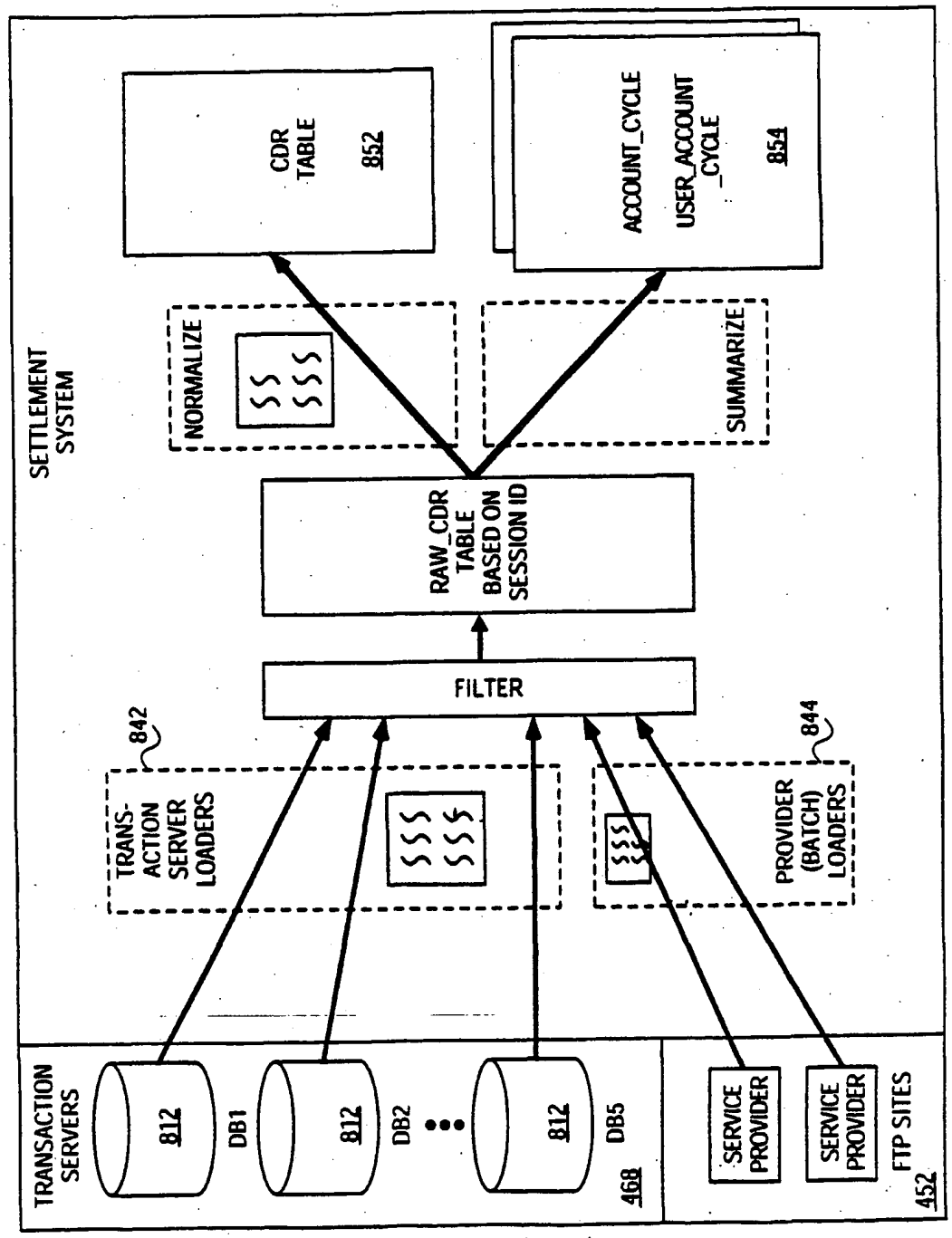

SETTLEMENT SYSTEM
CDR TABLE 852
ACCOUNT_CYCLE
USER_ACCOUNT_CYCLE 854
NORMALIZE
SUMMARIZE
RAW_CDR TABLE BASED ON SESSION ID
FILTER
842
TRANS-ACTION SERVER LOADERS
844
PROVIDER (BATCH) LOADERS
TRANSACTION SERVERS
812
DB1
812
DB2
812
DB5
468
SERVICE PROVIDER
SERVICE PROVIDER
FTP SITES
452

FIG. 17B

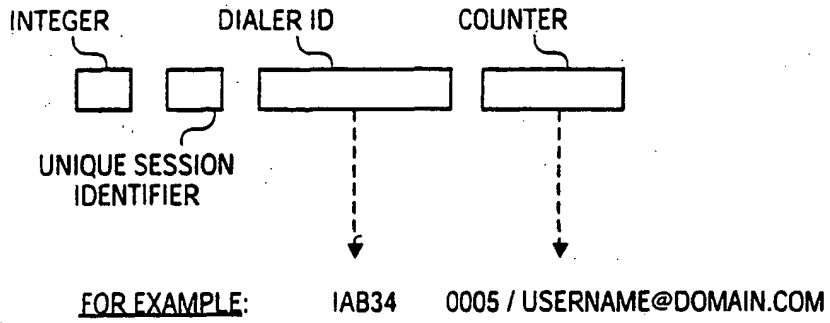

INTEGER
DIALER ID
COUNTER
UNIQUE SESSION IDENTIFIER
FOR EXAMPLE:
IAB34
0005 / USERNAME@DOMAIN.COM

FIG. 18

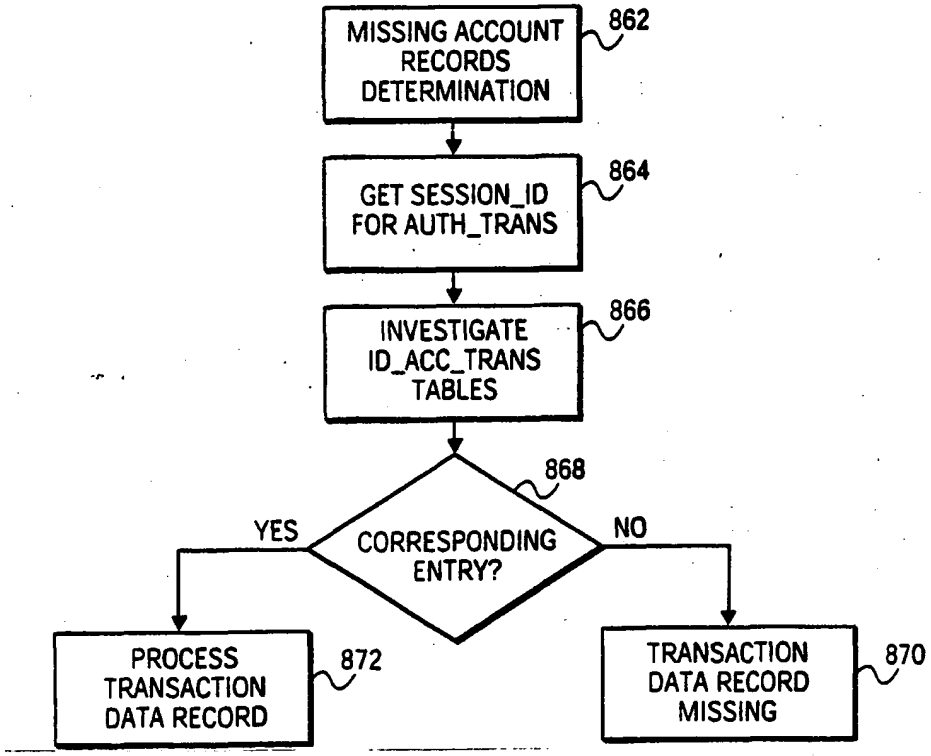

MISSING ACCOUNT RECORDS DETERMINATION
862
GET SESSION_ID FOR AUTH_TRANS
864
INVESTIGATE ID_ACC_TRANS TABLES
866
CORRESPONDING ENTRY?
868
YES
NO
PROCESS TRANSACTION DATA RECORD
872
TRANSACTION DATA RECORD MISSING
870

FIG. 19

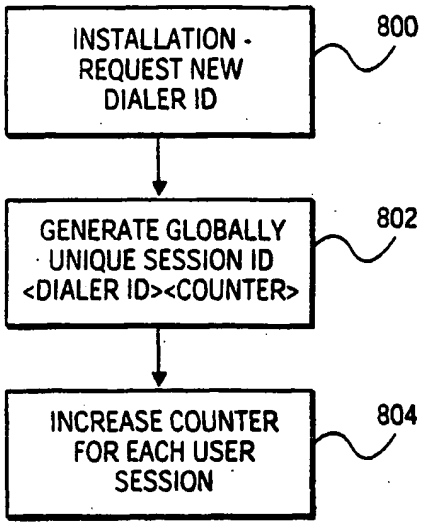

INSTALLATION - REQUEST NEW DIALER ID
800
GENERATE GLOBALLY UNIQUE SESSION ID <DIALER ID><COUNTER>
802
INCREASE COUNTER FOR EACH USER SESSION
804

FIG. 20

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US02/12470

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) : C06F 11/00; G06F 13/00; H04L 9/00, 9/08, 9/32
US CL : 380/277; 713/153, 155, 156, 158, 168, 181, 182, 202

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
U.S. : 380/277; 713/153, 155, 156, 158, 168, 181, 182, 202

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
WEST, INTERNET (IETF HOME PAGE for RFC's)- search terms: authentication, public key cryptography, credentials

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y, P	US 6,260,142 B1 (THAKKAR et al) 10 July 2001 (10.07.2001), column 3, line 26 to column 7, line 48.	1-46
Y	US 6,219,790 B1 (LLOYD et al) 17 April 2001 (17.04.2001), column 3, line 66 to column 14, line 39.	1-46
Y	US 6,198,824 B1 (SHAMBROOM) 06 March 2001 (06.03.2001), column 5, lines 12-57 and column 6, line 55 to column 9, line 50.	1-46
Y	US 6,189,096 B1 (HAVERTY) 13 February 2001 (13.02.2001), column 9, line 15 to column 13, line 10.	1-46
Y	US 5,497,421 A (KAUFMAN et al) 05 March 1996 (05.03.1996), column 3, line 66 to column 4, line 67 and column 5, line 22 to column 8, line 24.	1-46
Y	WO 97/15885 (ELLIS) 01 May 1997 (01.05.1997), page 3, line 23 to page 5, line 26 and page 14, line 34 to page 15, line 33.	1-46
Y	ABOBA, B. et al., Network Access Identifier, RFC2486, January 1999, pages 1-5.	11, 12, 20

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
28 August 2002 (28.08.2002)	16 SEP 2002
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703)305-3230	Authorized officer Matthew Smithers Peggy Harrod Telephone No. (703) 305-3900

Form PCT/ISA/210 (second sheet) (July 1998)